

**UNIVERSIDAD NACIONAL DE INGENIERÍA**



**FACULTAD DE ELECTROTECNIA Y COMPUTACION**

**INGENIERÍA ELECTRÓNICA**

**SISTEMA DE MONITOREO PREVENTIVO EN CASO DE INUNDACIÓN  
PLUVIAL EN EL DISTRITO V DE MANAGUA**

**TRABAJO MONOGRAFICO PARA OPTAR AL TITULO DE INGENIERO  
ELECTRONICO**

**ELABORADO POR:**

**BR. JORGE NOEL REAL MARTÍNEZ/ CARNET: 201361894**

**TUTOR: ING. ALVARO GAITAN**

**MANAGUA, NICARAGUA**

**MAYO 2019**

## **AGRADECIMIENTOS**

En primer lugar agradezco a la facultad por aprobar el protocolo monográfico que me permitió iniciar con este trabajo. De la misma manera sus buenos oficios para la gestión de solicitud de información en la alcaldía de Managua.

Seguidamente debo agradecer el apoyo de la Ing. Janixia Palma, de la Alcaldía de Managua por ayudarme a gestionar información importante para documentar este trabajo.

Así mismo, agradezco a mi tutor Ing. Álvaro Gaitán por corregir el documento monográfico, como en su apoyo profesional en ciertas dudas de carácter tecnológico que surgieron durante el desarrollo del mismo.

## **RESUMEN**

El presente documento monográfico pretende establecer un referente para el desarrollo de sistemas de monitoreo de nivel de agua pluvial en tiempo real abierto al público con acceso a internet. El documento incluye el proceso de diseño, implementación y análisis de resultados de un sistema prototipo con el cual el usuario por medio de una aplicación Android podrá monitorear en tiempo real el nivel de agua pluvial de una zona en particular durante un periodo lluvioso.

También este sistema supone una herramienta de apoyo a las instituciones que puedan mostrarse interesadas en obtener de forma gratuita estos importantes datos generados durante fenómenos lluviosos prolongados. Por tanto podría servir de apoyo a sistemas de alerta temprana.

En el ámbito tecnológico, especialmente en el internet de las cosas (IoT) y de las Tecnologías de la Información y la Comunicación (TIC) se emplea un dispositivo GSM/GPRS único para transmisión y recepción de datos con tecnología inalámbrica. Un servidor en línea captura los datos enviados por este módulo y los guarda en una base de datos, desde la cual son accedidos por nuestra aplicación Android.

## CONTENIDO

INTRODUCCION .....	1
OBJETIVOS .....	3
JUSTIFICACION .....	4
CAPITULO 1: MARCO TEORICO .....	5
1.1. INUNDACIONES .....	5
1.1.1. INUNDACION PLUVIAL .....	5
1.1.2. CRITERIOS PARA LA EVALUACION DE LA INTENSIDAD DE LA INUNDACION .....	6
1.2. UBICACIÓN DE LOS PUNTOS CRITICOS EN EL MUNICIPIO DE MANAGUA...	8
1.3. EL SENSOR DE NIVEL .....	9
1.3.1. EL SENSOR DE ULTRASONIDOS .....	10
1.4. EL MICROCONTROLADOR.....	11
1.4.1. CRITERIOS DE SELECCIÓN DEL MICROCONTROLADOR.....	12
1.5. TRANSMISION SERIE .....	13
1.6. EL MODULO USART/SCI EN EL PIC16F688.....	13
1.7. EL COMPILADOR CCS C .....	15
1.8. CONFIGURACION DEL MODULO USART EN C.....	16
1.8.1. INTERRUPCION POR RECEPCION DE DATOS EN C .....	17
1.9. LA NORMA DE COMUNICACIÓN RS232 .....	18
1.10. USO DEL TIMER1 COMO CONTADOR EN CCS C.....	19
1.11. EL MODULO GSM/GPRS .....	20
1.11.1. EL MODULO GSM/GPRS Y SU PUERTO SERIE .....	21
1.12. EL MODELO TCP/IP .....	23
1.12.1. EL PROTOCOLO HTTP .....	25
1.13. EL SERVIDOR WEB .....	27
1.14. EL SERVIDOR APACHE .....	27
1.15. EL HOSTING .....	28
1.16. PHP .....	28
1.17. MYSQL.....	29
1.18. MIT APP INVENTOR.....	29
1.19. ENERGIA SOLAR FOTOVOLTAICA .....	30
1.9.1. CELULAS SOLARES.....	31
1.19.1.1. LAS CELULAS DE SILICIO CRISTALINO .....	31

1.19.1.2. INCLINACION Y ORIENTACION DEL MODULO .....	31
1.19.2. CONTROLADOR DE CARGA Y ACUMULADOR EN SISTEMAS FOTOVOLTAICOS .....	32
CAPITULO 2: ANALISIS Y PRESENTACION DE RESULTADOS.....	34
2.1. ANALISIS DEL SISTEMA .....	34
2.1.1. ANALISIS DE FUENTES DE INFORMACION .....	34
2.1.2. CRITERIOS DE DISEÑO .....	39
2.2. DISEÑO DEL SISTEMA .....	40
2.2.1. ESTUDIO DE FACTIBILIDAD DE LOS PRINCIPALES COMPONENTES ...	40
2.2.1.1. SELECCIÓN DEL SENSOR DE NIVEL .....	41
2.2.1.2. SELECCIÓN DEL MICROCONTROLADOR .....	42
2.2.1.3. SELECCIÓN DE MODULO DE TRANSMISION .....	44
2.2.1.3.1. SELECCIÓN ENTRE DIVERSOS MODULOS GSM/GPRS .....	46
2.2.1.4. SELECCIÓN DEL SISTEMA DE ALIMENTACION .....	47
2.2.1.5. SELECCIÓN DEL SOFTWARE .....	48
2.2.1.5.1. SELECCIÓN DEL COMPILADOR PARA EL PIC .....	48
2.2.1.5.2. SELECCIÓN DEL SIMULADOR DE CIRCUITOS .....	49
2.2.1.5.3. SELECCIÓN DE LA HERRAMIENTA DE DESARROLLO DE APPS .....	49
2.2.1.5.4. SELECCIÓN DEL LENGUAJE DE PROGRAMACION WEB .....	50
2.2.1.5.5. SELECCION DEL GESTIONADOR DE BASES DE DATOS .....	50
2.2.1.6. SELECCION DEL TIPO DE SERVIDOR .....	51
2.2.2. DESCRIPCION DE SISTEMA .....	52
2.2.3. DISEÑO DEL SISTEMA DAQ.....	54
2.2.4. DISEÑO DEL SISTEMA DE COMUNICACIÓN Y TRANSMISION.....	56
2.2.4.1. SISTEMA DE COMUNICACION .....	56
2.2.5. DISEÑO DEL SISTEMA DE CONTROL .....	59
2.2.5.1. SIMULACION DEL SISTEMA DE COMUNICACIÓN Y CONTROL.....	61
2.2.5.2. COMANDOS PARA CONFIGURACION DE INTERNET.....	62
2.2.6. DISEÑO DEL SERVIDOR WEB .....	63
2.2.6.1. PROCESO DE SUBIDA DE DATOS AL SERVIDOR .....	64
2.2.6.2. PROCESO DE RECUPERACION DE DATOS.....	65
2.2.7. DISEÑO DE LA BASE DE DATOS .....	65
2.2.8. DISEÑO DE LOS SCRIPTS DEL SERVIDOR .....	67
2.2.9. DISEÑO DE LA APLICACIÓN ANDROID .....	73

2.2.9.1. SCREEN PRINCIPAL .....	73
2.2.9.2. SCREEN DE CONSULTAS .....	78
2.2.9.3. SCREEN DE ALERTAS.....	81
2.2.10. DISEÑO DEL SISTEMA FOTOVOLTAICO .....	82
2.2.10.1. CALCULO DE LA CAPACIDAD DE LA BATERIA.....	82
2.2.10.2. CALCULO DE LA CAPACIDAD DEL MODULO FOTOVOLTAICO .....	83
2.2.11. DISEÑO DE LAS PISTAS DE LA PCB .....	85
2.3. IMPLEMENTACION Y EVALUACION .....	87
2.3.1. PUESTA EN MARCHA DEL SISTEMA.....	88
2.3.2. EVALUACION DEL SISTEMA .....	91
REFERENCIAS BIBLIOGRAFICAS .....	93

## ANEXOS

### ANEXO A: ESPECIFICACIONES DE SENSORES, ICS Y MODULOS

#### A.1. SENSOR ULTRASONICO HC-SR04

#### A.2. DRIVER/RECEPTOR MAX232

#### A.3. MICROCONTROLADOR PIC16F688

##### A.3.1. DESCRIPCION GENERAL DEL PIC16F688

##### A.3.2. MODULO EUSART EN EL PIC16F688

###### A.3.2.1. MODO ASINCRONO EN EL EUSART

###### A.3.2.2. TRANSMISION ASINCRONA EN EL EUSART

###### A.3.2.3. RECEPCION ASINCRONA DEL EUSART

###### A.3.2.4. EL GENERADOR DE BAUDRATE (BRG) DEL EUSART

##### A.3.3. MODULO TIMER1

###### A.3.3.1. OPERACIÓN DEL TIMER1

###### A.3.3.2. SELECCIÓN DE FUENTE DE RELOJ

###### A.3.3.3. PRESCALER DEL TIMER1

###### A.3.3.4. OSCILADOR DEL TIMER1

###### A.3.3.5. REGISTRO DE CONTROL DEL TIMER1

#### A.4. MODULO GSM/GPRS SIM900

##### A.4.1. DESCRIPCION GENERAL DEL SIM900

###### A.4.1.1. CARACTERISTICAS CLAVES DEL SIM900

###### A.4.1.2. MODOS DE OPERACION

###### A.4.1.3. DIAGRAMA FUNCIONAL DEL SIM900

###### A.4.1.4. DESCRIPCION DE PINES

### ANEXO B: COMANDOS AT DEL SIM900 VERSION 1.07

B.1. COMANDOS AT PARA LA APLICACIÓN IP

B.2. COMANDOS AT PARA LA APLICACIÓN HTTP

ANEXO C: DIRECTIVAS Y FUNCIONES PREDEFINIDAS EN CCS

C.1. PARAMETROS DE CONFIGURACION DEL USART

C.2. FUNCIONES PREDEFINIDAS ASOCIADAS AL USART

ANEXO D: CODIGOS DE PROGRAMACION

D.1. CODIGO DEL MICROCONTROLADOR PIC 16F688

D.2. CODIGO PHP DEL SCRIPT 1

D.3. CODIGO PHP DEL SCRIPT 2

D.4. CODIGO PHP DEL SCRIPT 3

D.5. CODIGO PHP DEL SCRIPT 4

ANEXO E: CODIGOS DE BLOQUE DE MIT APP INVENTOR

E.1. CODIGO DE SCREEN PRINCIPAL

E.2. CODIGO DE SCREEN CONSULTAS

E.3. CODIGO DE SCREEN ALERTAS

ANEXO F: CREACION DEL SERVIDOR WEB Y LA BASE DE DATOS

F.1. PASOS PARA MONTAR EL SERVIDOR CON UN WEB HOSTING

F.2.3 PASOS PARA LA CREACION DEL LA BASE DE DATOS

ANEXO G: ENCUESTA

ANEXO H: PRESUPUESTO

## INTRODUCCIÓN

El tema del cambio climático, elementos sensores y tecnología para transmisión de datos son de especial interés para el desarrollo de sistemas de monitoreo en tiempo real que a la vez puede ser parte de un sistema de alerta temprana (SAT) contra inundaciones, lo que resulta en una apropiada gestión del riesgo.

El cambio climático, atribuido a un cambio natural como también a los resultados de la influencia del ser humano, es un reto enorme ya que sus efectos son variados, casi siempre negativos y difícil de manejar. Entre estos efectos podemos mencionar las sequías, huracanes e inundaciones (Conde & Saldaña, 2007).

En el caso particular de las inundaciones es necesario conocer entonces algunos parámetros a medirse, como la precipitación o el nivel de agua, mediante un sistema de adquisición de datos y que estos datos sean transmitidos en tiempo real hacia una central (servidor) donde serán almacenados para su posterior análisis. Los datos obtenidos de los sensores son importantes a la hora de tomar decisiones

Con el desarrollo tecnológico actual es posible montar sistemas con las características descritas anteriormente y a un precio relativamente bajo. De esta forma la necesidad de información respecto a la obtención de datos en tiempo real quedaría cubierta con el claro objetivo de análisis del entorno y la toma temprana de decisiones. Con esta información se pretende que las autoridades pertinentes realicen una apropiada gestión del riesgo, que son aquellas actividades destinadas a minimizar los efectos de los desastres, incluyendo efectos posteriores como por ejemplo la rehabilitación y reconstrucción del territorio (Domínguez & Lozano, 2014 ).

El municipio de Managua es altamente vulnerable a las inundaciones pluviales ya que el sistema pluvial no es eficiente en cuanto a la capacidad de drenaje de las



aguas pluviales, esto debido a los sedimentos de los cauces, cuencas y afluentes y esto sin contar con la basura que reduce la efectividad de los canales de drenaje. Así hay muchos puntos dentro de la zona de Managua llamados puntos críticos que se tornan vulnerables ante la amenaza de inundación pluvial. (Amec Foster Wheeler, 2017).

En el entorno nicaragüense, INETER tiene un sistema de vigilancia que consiste en una red telemétrica para el monitoreo de varios parámetros, pero particularmente para Managua y otras regiones no cuentan con un sistema que esté destinado a medir nivel de agua pluvial en los puntos críticos del municipio de Managua. Aunque cuentan con una red hidroluviotelemétrica auspiciado por la alcaldía de Managua, este sistema mide pluviosidad que representa la cantidad de lluvia en litros, pero esto no es lo mismo que medir nivel de agua. Además, estos sistemas no están ubicados en los puntos críticos sino cerca de las principales microcuencas. (Amec Foster Wheeler, 2017).

.Como podrá comprenderse el propósito de este trabajo es desarrollar un sistema, a nivel de prototipo, que pretenda satisfacer las necesidades de información de nivel de agua en tiempo real para la toma de decisiones durante un fenómeno prolongado de lluvia. Se debe aclarar que debido al alcance económico del proyecto este solo podría monitorear un punto crítico en la zona del municipio de Managua. Sin embargo, este se diseñará como un sistema escalable y abierto haciendo que se puedan agregar más dispositivos de medición y transmisión a la red.

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Desarrollar un sistema de monitoreo preventivo en tiempo real en caso de inundación pluvial para lograr una adecuada gestión del riesgo en el distrito V del municipio de Managua.

### **OBJETIVOS ESPECÍFICOS**

- ✓ Identificar el lugar específico en donde ubicar el sensor para obtener datos acerca del nivel de agua pluvial.
- ✓ Diseñar el sistema de adquisición de datos para medir profundidad de agua pluvial.
- ✓ Implementar un sistema de transmisión de datos remoto con tecnología inalámbrica usando una de las redes telefónicas disponibles.
- ✓ Crear una base de datos en donde se pueda consultar el registro de los datos y el historial de alertas.
- ✓ Desarrollar una aplicación Android en donde se publiquen los datos de nivel de agua y las alertas emitidas por el sistema en tiempo real.

## JUSTIFICACION

En el distrito V de Managua existen puntos críticos de inundación y a pesar de esto no se ha desarrollado algún tipo de sistema de monitoreo. El desarrollo de un sistema de monitoreo en esta zona, sería además de novedoso en la gestión del riesgo de inundación, una importante herramienta en la toma de decisiones para las autoridades involucradas en rescate. Esta toma de decisiones consiste principalmente en que las autoridades pertinentes (SINAPRED por ejemplo), puedan desplegar a tiempo y más rápido las brigadas de rescate y salvamento para evacuar la zona en caso que una inundación o anegamiento ponga en peligro la integridad física de las personas. Aunque en general se pretende que sea un sistema abierto al público.

También el aporte está desde el punto de vista del servicio de monitoreo. Este servicio aunque no será una novedad como tecnología, si lo es en la forma que se implementará, ya que se hará uso de tecnología inalámbrica para transmisión con dispositivos Android como medio de visualización en tiempo real. No se tiene conocimiento o precedente que se haya desarrollado o montado un sistema real en Nicaragua con propósitos de monitoreo de inundación.

Desde la perspectiva profesional, la creación de este sistema permitirá obtener habilidades en el desarrollo de servicios TIC, como son el desarrollo de una aplicación Android, instalación de un servidor web, así como también una base de datos asociada a la aplicación. También como parte de las TIC, el sistema de telecomunicación que se implementará implica el aprendizaje de comandos necesarios para establecer la transmisión y recepción usando la red telefónica local como canal de comunicación.

## **CAPITULO 1: MARCO TEORICO**

### **1.1. INUNDACIONES**

Las inundaciones se producen cuando lluvias intensas o continuas sobrepasan la capacidad de retención e infiltración del suelo, la capacidad máxima de transporte del río o arroyo es superada y el cauce principal se desborda e inunda los terrenos cercanos a los propios cursos de agua. (INETER-COSUDE, 2005).

Las inundaciones pueden clasificarse según su duración y mecanismo de generación.

Según su duración pueden ser:

- Inundaciones rápidas o dinámicas.
- Inundaciones lentas o estáticas.

Según el mecanismo de generación:

- Inundaciones pluviales.
- Inundaciones fluviales.

#### **1.1.1. INUNDACION PLUVIAL**

Una inundación pluvial es la que se produce por la acumulación de agua de lluvia en un determinado lugar o área geográfica sin que ese fenómeno coincida necesariamente con el desbordamiento de un cauce fluvial. Este tipo de inundación se genera tras un régimen de precipitaciones intensas o persistentes, es decir, por la concentración de un elevado volumen de lluvia en un intervalo de tiempo muy breve o por la incidencia de una precipitación moderada y persistente durante un amplio período de tiempo sobre un suelo poco permeable (INETER-COSUDE, 2005).

### 1.1.2. CRITERIOS PARA LA EVALUACIÓN DE LA INTENSIDAD DE LA INUNDACION

Los criterios recomendados para evaluar la intensidad de las inundaciones son diferentes en dependencia del tipo de inundación como se puede ver en la siguiente tabla:

**Tabla 1.1**

*Criterio de evaluación de la magnitud de la inundación*

Tipo de inundación	Criterio de evaluación
Estática	Profundidad o altura del flujo
Dinámica	Producto de la velocidad por la profundidad del flujo

**Nota.** Fuente: INETER-COSUDE. (2005). Inundaciones fluviales. Mangua, Nicaragua: INETER-COSUDE.

Los umbrales entre los niveles de intensidad alta, media y baja, han sido definidos considerando la peligrosidad que una determinada columna de agua puede significar para la infraestructura o las viviendas y la vida de los pobladores. (INETER-COSUDE, 2005).

En la figura 1.1 se presentan los rangos definidos para cada nivel de intensidad para inundaciones.

Niveles de intensidad	Profundidad del flujo (H) (m) (Inundaciones estáticas)	Profundidad x vol. del flujo (m <sup>2</sup> /s) (Inundaciones dinámicas)
Alto	$H \geq 1.0$	$H*V \geq 1.5$
Medio	$0.5 \leq H < 1.0$	$0.5 \leq H*V < 1.5$
Bajo	$0.25 \leq H < 0.5$	$H*V < 0.5$ y $H \geq 0.25$ m

Inundaciones Estáticas		
Profundidad	$\geq 1.00$ m	
	0.5 - 1.00 m	
	0.25 - 0.50 m	

Inundaciones Dinámicas				
Profundidad	1.50 m	0.75	1.50	2.25
	1.00 m	0.50	1.00	1.50
	0.50 m	0.25	0.50	0.75
		0.50	1.00	1.50
		Velocidad (m/s)		

#### Leyenda de Colores

	Intensidad Alta
	Intensidad Media
	Intensidad Baja

**Figura 1.1.** Rangos para cada nivel de intensidad de inundación.

Fuente: INETER-COSUDE. (2005). Inundaciones fluviales. Mangua, Nicaragua: INETER-COSUDE.

Las **inundaciones de alta intensidad** corresponden a aquellas que presentan profundidades de flujo mayores a 1m o el producto resultante de la velocidad por altura ( $V*H$ ) es mayor a 1.5 m<sup>2</sup>/s. Los daños causados por una inundación de alta intensidad generalmente son altos en pérdidas de vidas y para la economía.

Las **inundaciones de media intensidad** son aquellas con altura (H) de agua entre 0.5 y 1m o el producto resultante de la velocidad por altura ( $V*H$ ) entre 0.5 y 1.5m<sup>2</sup>/s. Los daños económicos y a la población son menores que en el caso de la inundación intensa, pero no despreciables.

Las **inundaciones de baja intensidad** corresponden a aquellas con profundidad del flujo superiores a 0.25m pero inferiores a los 0.5m, o  $V*H$  menor a 0.5 m<sup>2</sup>/s. Los daños asociados son generalmente leves, no se esperan pérdidas en vidas humanas. (INETER-COSUDE, 2005).

## **1.2. UBICACIÓN DE LOS PUNTOS CRÍTICOS EN EL MUNICIPIO DE MANGUA**

Por el contexto del PMES<sup>1</sup>, un punto crítico es un lugar físico o zona que es de alto riesgo para la seguridad de los habitantes de dicha zona, dadas las condiciones de dicha zona en situación de intensos regímenes de lluvia. En periodos lluviosos intensos y sistemáticos y en dependencia de la ubicación de un punto crítico, puede ocurrir lo siguiente:

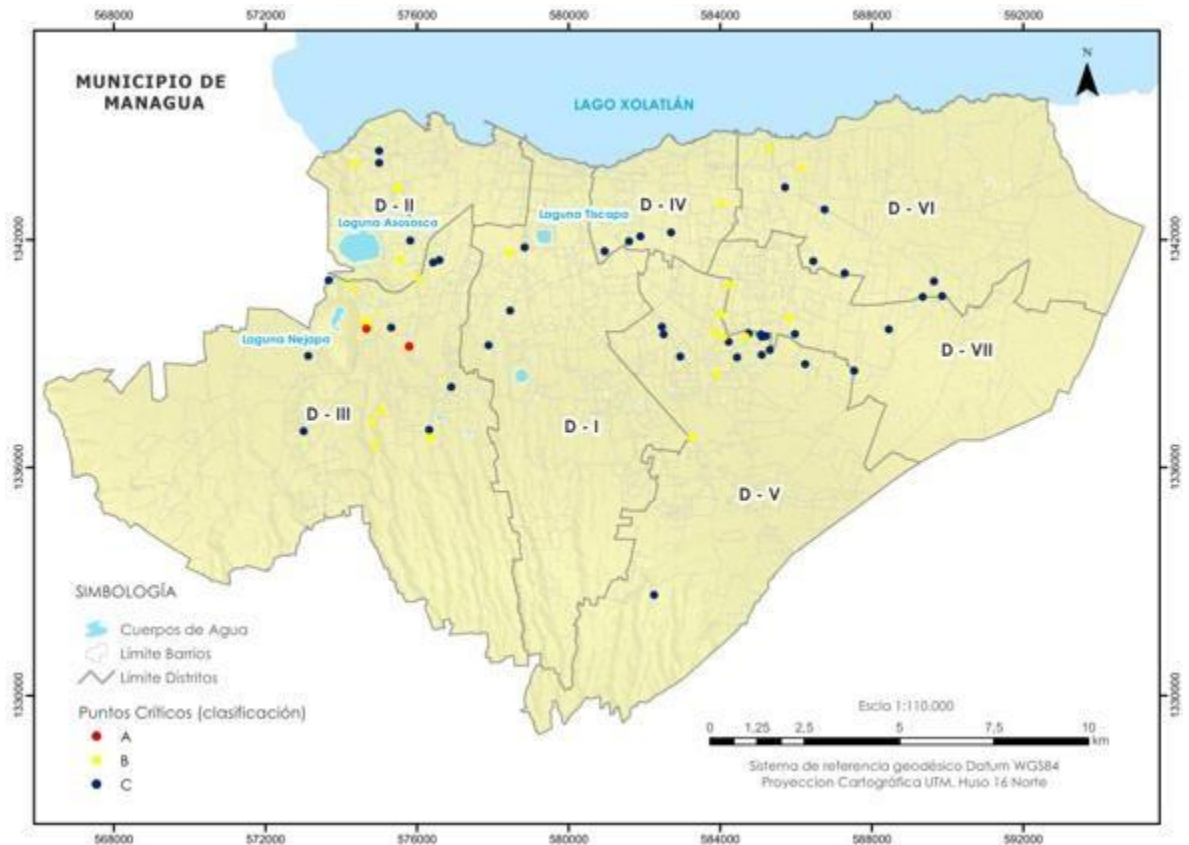
- Inundación
- Derrumbe
- Anegación temporal (inundación temporal), En viviendas y en estas se encuentran menores de edad y personas discapacitadas.
- Crecimiento del nivel del lago (zonas cercanas al lago.)

De lo anterior, nos vamos a centrar en puntos críticos de inundación pluvial y más específicamente el anegamiento, el cual hace su aparición durante fenómenos lluviosos intensos y prolongados. (Amec Foster Wheeler, 2017)

En la figura 1.2 podemos observar la distribución de los puntos críticos de inundación pluvial ubicados en el municipio de Managua.

---

<sup>1</sup> PMES: Plan Maestro de Escorrentía Superficial. Elaborado por Amec Foster Wheeler.



**Figura 1.2.** Distribución de los puntos críticos en el municipio de Managua

Fuente: Amec Foster Wheeler. (2017). PLAN MAESTRO DE ESCORRENTIA SUPERFICIAL (PMES) DEL MUNICIPIO DE MANAGUA. Miami, FI.

### 1.3. EL SENSOR DE NIVEL

Para realizar mediciones de nivel de líquidos existen varios métodos usados en automatización como por ejemplo se puede emplear un flotador con una conexión mecánica que, en virtud del principio de Arquímedes, convierte el desplazamiento de la superficie libre en una fuerza o par, y estos en ángulos de giro. Otra posibilidad es emplear un flotador con una polea y un contrapeso, donde el ángulo girado por la polea es proporcional al nivel del líquido. (Pallás, 2003).

Así también existen otros métodos como medir la diferencia de presión en el fondo del depósito y a partir de ahí calcular la altura del líquido o con el método del burbujeo. (Pallás, 2003).



Refiriéndonos en particular a los instrumentos disponibles para realizar las mediciones de nivel de líquidos, existe una amplia variedad. Entre estos tenemos las varillas graduadas (dipsticks), sistemas flotantes, sistemas hidrostáticos o de medición de presión y capacitivos. (Morris, 2001)

También existen otras tecnologías modernas sin contacto para realizar dichas mediciones como indicador ultrasónico de nivel, instrumentos de ondas de radar, de radiación, de láser, de fibra óptica, de termografía e incluso instrumentos inteligentes de medición que cuentan con microprocesadores para simplificar la instalación. (Morris, 2001)

### 1.3.1. EL SENSOR DE ULTRASONIDOS HC-SR04

Se pondrá especial énfasis en el sensor ultrasónico<sup>2</sup> (fig. 1.3), ya que cuenta con características técnicas representativas de un sensor dedicado a la medición de nivel. Aunque en el capítulo siguiente se dará una descripción más detallada del mismo, se explicará un poco el funcionamiento para tener una idea del uso de este sensor.



**Figura 1.3.** Sensor ultrasónico HC-SR04

Fuente: <https://www.geekfactory.mx/tienda/sensores/hc-sr04-sensor-de-distancia-ultrasonico/>

Funcionamiento:

Primero se debe inyectar un pulso en el pin Trig, por al menos 10us. El modulo automáticamente enviará un tren de ocho pulsos ultrasónicos de 40kHz y

---

<sup>2</sup> Ver datasheet en Anexos A.1

detectará si hay un pulso de señal de retorno. Si hay señal de retorno en el pin Echo se generará un pulso con la duración medida desde que se generó el tren de pulsos hasta que su eco ha sido recibido por el modulo. Es decir el modulo pone en estado alto el pin Echo por un tiempo (high level time) igual a tiempo que le tomo generar el tren de pulsos ultrasónicos hasta fue recibido su eco.

#### **1.4. EL MICROCONTROLADOR**

El microcontrolador es un circuito integrado programable. En su memoria sólo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar, y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada. (Angulo, 2003).

Un microcontrolador tiene un CPU (un microprocesador) además de una fija cantidad de RAM, ROM, puertos I/O y un timer todo en un simple chip. En otras palabras, el procesador, RAM, ROM, puertos I/O, y timer están embebidos juntos en un chip; por lo tanto, el diseñador no puede agregar memoria externa, I/O, o timer a él. La cantidad fija sobre el chip de ROM, RAM, y número de puertos I/O en los microcontroladores los hace ideal para muchas aplicaciones en los cuales el costo y el espacio son críticos. (Ali, 2008).

Existen cinco microcontroladores principales de 8 bits. Estos son: de Freescale Semiconductor (antes Motorola) 68HC08/68HC11, de Intel 8051, de Atmel AVR, de Zilog Z8, y PIC de Microchip Technology. Cada uno cuenta con su propio set de instrucciones y registros, por lo tanto no son compatibles entre sí. Los programas escritos en unos no correrán en otros. También existen microcontroladores de 16 bits y 32 bits. (Ali, 2008). En la figura 1.4 observamos una ilustración de un típico microcontrolador en su versión de 8 bits.



**Figura 1.4.** Microcontrolador Atmel Atmega328P

Fuente: <https://www.lelong.com.my/atmel-atmega328p-8-bit-avr-microcontroller-dip-28-littlecraft-185724193-2019-11-Sale-P.htm>

#### **1.4.1. CRITERIOS DE SELECCIÓN DEL MICROCONTROLADOR**

Para la selección del microcontrolador se deben tomar en cuenta varios criterios. (Ali, 2008).

- 1) El primero y más importante criterio en la elección del microcontrolador es que debe cumplir con la tarea de forma eficiente y económicamente. En dependencia de las necesidades del proyecto debemos ver si un microcontrolador de 8,16 o 32 bits puede cumplir efectivamente con las necesidades de computación. Otra consideración en esta categoría son la velocidad, encapsulado, potencia de consumo, cantidad de RAM y ROM, numero de pines I/O y timer, fácil de actualizar a versiones de mayor desempeño o menor consumo de potencia y costo por unidad.
- 2) El segundo criterio en la elección es que tan fácil es desarrollar productos en torno al él. Aquí nos referimos a la disponibilidad de un assembler, debugger, un código eficiente en un compilador de lenguaje C, emulador y soporte técnico.
- 3) El tercer criterio para elegir un microcontrolador es su disponibilidad inmediata en cantidades necesarias tanto ahora como en el futuro.

## 1.5. TRANSMISION SERIE

Los PIC utilizan, entre otros, dos modos de transmisión en serie:

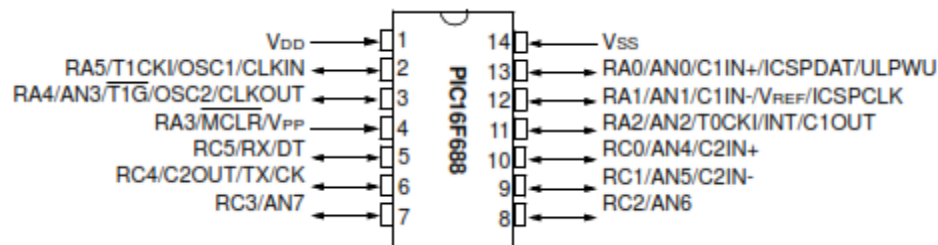
- El puerto serie síncrono (SSP)
- La interfaz de comunicación serie (SCI) o receptor transmisión serie síncrono universal (USART).

La comunicación USART (transmisor-receptor serie síncrono-asíncrono universal), también conocido como SCI (interfaz de comunicación serie), permite la comunicación con un ordenador trabajando en modo full-duplex asíncrono o con periféricos trabajando en modo half-duplex. En general, puede trabajar de dos formas.

- Asíncrono (full dúplex).
- Síncrono (half-duplex). (García, 2008).

## 1.6. EL MODULO USART/SCI EN EL PIC16F688

Muchos microcontroladores cuentan con un módulo USART el cual permite realizar comunicación serie usando uno de los protocolos de los cuales tiene soporte. En la figura 1.5 se muestra el diagrama de pines de un microcontrolador. El módulo USART opera, para este microcontrolador, a través de los pines RX (pin5) y TX (pin6).



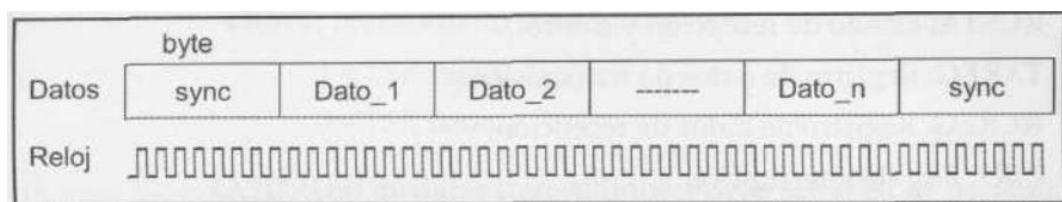
**Figura 1.5.** Microcontrolador PIC16F688 y su diagrama de pines.

Fuente: Data sheet del PIC16F688

La principal función del USART es la de transmitir o recibir datos en serie. Esta operación puede dividirse en dos categorías: síncrona y asíncrona. La transmisión síncrona utiliza una señal de reloj y una línea de datos, mientras que en la transmisión asíncrona no se envía señal de reloj, por lo que el emisor y el receptor deben tener relojes con la misma frecuencia y fase. Cuando la distancia entre el emisor y el receptor es pequeña se suele utilizar transmisión síncrona, mientras que para distancias mayores se utiliza la transmisión asíncrona. (García, 2008).

El USART puede transmitir o recibir datos serie a una velocidad en baudios (bits/s). Puede transferir tramas de datos de 8 o 9 bits por transmisión y detectar errores de transmisión. También puede generar interrupciones cuando se produce una recepción de datos o cuando la transmisión ha sido completada. (García, 2008).

En el modo síncrono se permite la transmisión de datos y no existe un límite de tamaño, es un modo semi-duplex (la comunicación serie se establece a través de una única línea en ambos sentidos, pero no se puede transferir información en ambos sentidos de forma simultanea). (García, 2008). En la figura 1.6 se muestra un diagrama de tiempo del modelo de transmisión síncrona.

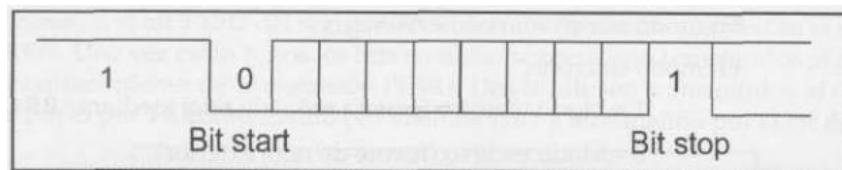


**Figura 1.6.** Transmisión síncrona

Fuente: García, E. (2008). Compilador C CCS y simulador PROTEUS para Microcontroladores PIC (1ra ed.). México D.F, México: Alfaomega

En el modo asíncrono se emplean relojes tanto en el emisor como en el receptor. Ambos relojes deben ser de igual frecuencia y deben estar en fase o sincronizados. La frecuencia de reloj se acuerda antes de la transmisión configurando la velocidad mientras que la sincronización se realiza durante la

transmisión. Cada trama de datos tiene un tamaño fijo y poseen un bit inicial de arranque (start) y un bit final o de parada (stop) que permiten realizar dicha sincronización. La transmisión es modo full-duplex (se utilizan dos líneas, una transmisora -TX- y otra receptora -RX-, transfiriendo información en ambos sentidos; se puede transmitir y recibir información de forma simultanea).



**Figura 1.7.** Transmisión asíncrona

Fuente: García, E. (2008). Compilador C CCS y simulador PROTEUS para Microcontroladores PIC (1ra ed.). México D.F, México: Alfaomega

El modo más habitual de transmisión por el USART es el modo asíncrono, puesto que permite comunicaciones en largas distancias. Existen distintas normas de transmisión serie asíncrona, como la RS232, la RS485, etc. Los niveles de tensión empleado en estas normas son diferentes al empleado por el PIC. Por ello, suele ser necesaria la utilización de circuitos externos de adaptación. (García. 2008)

Para usar el módulo USART del PIC de la figura 1.5, se deben configurar los pines RC5/RX/DT (pin 5) y RC4/C2OUT/TX/CK (pin 6), así mismo el propio módulo USART.

## 1.7. EL COMPILADOR CCS C

El compilador C de CCS ha sido desarrollado específicamente para PIC MCU, obteniendo la máxima optimización del compilador con estos dispositivos. Dispone de una amplia librería de funciones predefinidas, comandos de procesado y ejemplos. Además, suministra los controladores para diversos dispositivos como LCD, convertidores AD, relojes en tiempo real, EEPROM serie, etc.

El compilador convierte el lenguaje de alto nivel a instrucciones en código maquina (ensamblador); un cross-compiler es un compilador que funciona en un procesador (normalmente en un PC) diferente al procesador objeto. El compilador CCS C es un cross-compiler. Los programas son editados y compilados a instrucciones maquina en el entorno de trabajo del PC, el código maquina puede ser cargado del PC al sistema PIC mediante el ICD2 (o mediante cualquier programador) y puede ser depurado (puntos de ruptura, paso a paso, etc.) desde el entorno de trabajo del PC.

El CCS C es C estándar y, además de las directivas estándar (#include, etc.), suministra unas directivas específicas para PIC (#device, etc.); además incluye funciones específicas (bit\_set(), etc.). Se suministra un editor que permite controlar la sintaxis del programa. (García, 2008).

## 1.8. CONFIGURACION DEL MODULO USART EN C

El compilador CCS C cuenta con la directiva **#USE RS232 (opciones)** que sirve para configurar varios parámetros del USART. Del mismo modo, el compilador cuenta con funciones predefinidas en sus directivas que permite realizar la transmisión y recepción de los datos a través del bus serie. Esta directiva y funciones asociadas permiten configurar los registros a nivel de bits en transmisión y en recepción, haciendo que su uso sea relativamente simple en un lenguaje de alto nivel. (García, 2008).

La directiva #USE RS232, como se dijo maneja varios modos u opciones ingresadas a través de software, en anexos se pueden observar todos los modos.

**Tabla 1.2**

*Funciones asociadas a la transmisión y recepción USART*

Función	Descripción
---------	-------------

setup_uart(baud)	baud es una constante que define la velocidad. Un 1 enciende el USART y un 0 lo apaga. Con cualquier valor de velocidad, el USART se enciende.
set_uart_speed(baud)	Idéntica a la función anterior
putc(dato)	dato es un carácter de 8 bits. Esta función envía un carácter mediante la patilla XMIT.
putchar(dato)	
puts(string)	string es una cadena de caracteres constante o matriz de caracteres terminada con un 0. Manda los caracteres de la cadena, uno a uno, a través del bus RS 232 utilizando la función PUTC(); detrás de la cadena envía un RETURN (13) y un retorno de carro (10)
printf(fnombre, string, valores...)	El parámetro string es una cadena de caracteres constante o matriz de caracteres terminada con un 0. El parámetro fnombre son las funciones a utilizar para escribir la cadena indicada. Los parámetros valores son los valores a incluir en la cadena separados por comas.
getc()	Función que espera un carácter por la línea RS 232 y devuelve su valor.
kbhit()	Devuelve FALSE si getc() debe esperar a que llegue un carácter o TRUE si ya hay un carácter listo para ser leído por getc().

**Nota.** Fuente: García, E. (2008). *Compilador C CCS y simulador PROTEUS para Microcontroladores PIC* (1ra ed.). México D.F, México: Alfaomega.

### 1.8.1. INTERRUPCION POR RECEPCION DE DATOS EN C

Las interrupciones permiten a cualquier suceso interior o exterior interrumpir la ejecución del programa principal en cualquier momento. En el momento de producirse la interrupción, el PIC ejecuta un salto a la rutina de atención a la interrupción, previamente definida por el programador, donde se atenderá a la demanda de la interrupción. Cuando se termina de ejecutar dicha rutina, el PIC retorna a la ejecución del programa principal en la misma posición de la memoria de programa donde se produjo la interrupción. (García, 2003)

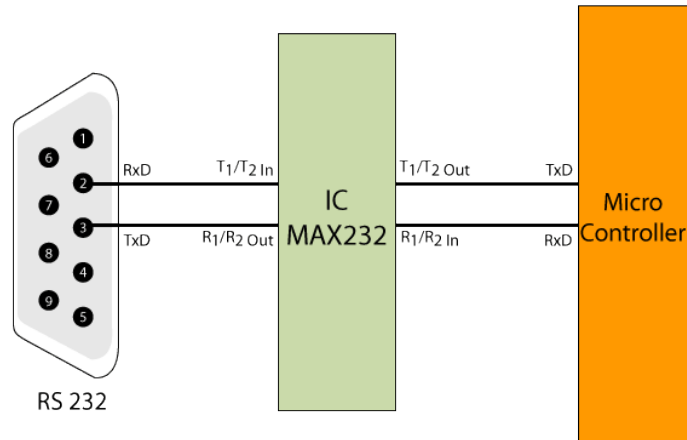


Para el uso de interrupciones por recepción de datos se debe incluir en el programa la directiva del compilador **#INT\_RDA**. Esta directiva debe ser definida antes de la función principal del programa en CCS C. Inmediatamente de la directiva se debe definir una función que responda a la interrupción. Dicha función no devuelve ningún valor (void), nada más realiza el proceso requerido por el programador durante la interrupción para luego devolver el control al programa principal.

Para habilitar la interrupción por recepción de datos con el compilador de C se usa **“enable\_interrupts(INT\_RDA);”**. Seguido a esto y para poder usar esta interrupción se deben habilitar todas la interrupciones agregando **“enable\_interrupts(GLOBAL);”** al código del programa. (García, 2003).

## **1.9. LA NORMA DE COMUNICACIÓN RS232**

Para permitir la compatibilidad entre equipos de comunicación de datos hechos por varios fabricantes, fue establecida una interface estándar llamada RS232 por Electronics Industries Association (EIA). Este estándar se usa en PCs y numerosos tipos de equipos. Debido a que la norma se estableció mucho antes de la llegada de la familia lógica TTL, sin embargo, sus niveles de voltaje de entrada y salida no son compatibles con TTL. En RS232, un 1 se representa de -3 a -25V, mientras que un bit 0 es +3 a +25V, donde de -3 a +3 es un estado indefinido. Por esta razón, para conectar cualquier sistema RS232 a un microcontrolador debemos usar convertidores de voltaje como el MAX232 para convertir los niveles lógicos TTL a niveles de voltaje RS232. (Ali, 2008).



**Figura 1.8.** Conexión básica full dúplex entre PIC y PC

Fuente: <https://www.engineersgarage.com/microcontroller/8051/projects/interface-serialport-RS232-AT89C51-circuit>

En la figura 1.8 se observa la conexión básica entre un sistema RS232 y un PIC, en donde el IC MAX232 actúa como interface entre ellos. En la figura, el sistema RS232 tiene un puerto o conector de 9 pines o DB9. Hoy en día, sin embargo, han ido desapareciendo los puertos serie de los PCs personales. Como solución se pueden utilizar módulos de conversión SERIE-USB que utilizan el Universal Serial Bus (USB), no se debe confundir con el modulo integrado USB con el que cuentan algunos PICs. (García, 2003).

## 1.10. USO DEL TIMER1 COMO CONTADOR EN CCS C

Los TIMER o temporizadores son módulos integrados en el PIC que permiten realizar cuentas tanto de eventos internos como externos. Cuando la cuenta es interna se habla de temporización y cuando es externa se habla de contador. El timer1 es un temporizador/contador que trabaja con 16 bits<sup>3</sup> y se puede leer y escribir él, entre otras características.

<sup>3</sup> Ver Anexo A.3.3.5 para más detalles sobre sus registros asociados

La configuración del módulo TIMER1<sup>4</sup> en el compilador de C se realiza a través de la función “**setup\_timer1(modos)**”.

La lectura y escritura en el módulo timer1 se realiza a través de las siguientes funciones “**valor=get\_timer1()**” y “**set\_timer1(valor)**”, donde valor es un entero de 16 bits.

### 1.11. EL MODULO GSM/GPRS

El módulo GSM/GPRS SIM900, mostrado en la figura 1.9, es un shield para las plataformas de arduino UNO o sus clones, para realizar comunicación inalámbrica usando las redes telefónicas disponibles. Pero también se puede comunicar con cualquier otro dispositivo que tenga un puerto de comunicación serie. Por ejemplo si se dispone de un cable SERIE-USB o un convertidor TTL a USB es posible comunicarlo con una PC o con un microcontrolador que cuente con un módulo USART.



**Figura 1.9.** Shield GSM/GPRS SIM 900

Fuente: <https://hetpro-store.com/gprs-simcom-sim900-quad-band-gsm-shield/>

El corazón de este shield es el chip SIM900<sup>5</sup> cuyas características generales son:

- Permite enviar y recibir SMS, MMS, GPRS y audio vía UART usando comandos AT

---

<sup>4</sup> Ver en Anexo C.3 todos los modos de configuración del TIMER1

<sup>5</sup> Ver detalles en Anexo A.4

- Tiene 12 GPIOs, 2 PWM y un ADC.
- Opera en cuatro frecuencias: GSM 850 MHz, EGSM 900 MHz, DCS 1800 MHz y PCS 1900 MHz, de forma que puede operar en muchos países con redes GSM.
- Control vía comandos AT
- Soporte para RTC (Reloj en tiempo real)
- Tiene integrado el protocolo TCP/IP; los comando AT TCP/IP extendidos se han desarrollado para clientes para usar el protocolo TCP/IP fácilmente, lo que es muy útil para aquellas aplicaciones de transferencia de datos.

#### **1.11.1. EL MODULO GSM/GPRS Y SU PUERTO SERIE.**

Antes de hablar de GPRS, nos referiremos a GSM (Global System Mobile), el cual es un estándar internacional de comunicaciones digitales móviles. Con este estándar diseñado principalmente para la transmisión de voz, pero también permite la transmisión de datos (SMS, internet) pero a una velocidad de 9.6 kb/s.

Debido a la limitación de velocidad de GSM se introdujo una evolución de GSM, GPRS (Global Packet Radio Service), basada en la transmisión por paquetes que ofrece servicios más avanzados, como una comunicación más eficiente, especialmente la comunicación con internet. (Sánchez, 2005)

En la figura 1.10 se muestra las características de conexión mediante el módulo GPRS del SIM900.

DATA GPRS:	<ul style="list-style-type: none"> <li>● GPRS data downlink transfer: max. 85.6 kbps</li> <li>● GPRS data uplink transfer: max. 42.8 kbps</li> <li>● Coding scheme: CS-1, CS-2, CS-3 and CS-4</li> <li>● Supports the protocols PAP (Password Authentication Protocol) usually used for PPP connections.</li> <li>● Integrates the TCP/IP protocol.</li> <li>● Support Packet Switched Broadcast Control Channel (PBCCH)</li> </ul>
CSD:	<ul style="list-style-type: none"> <li>● CSD transmission rates: 2.4, 4.8, 9.6, 14.4 kbps, non-transparent</li> <li>● Unstructured Supplementary Services Data (USSD) support</li> </ul>

**Figura 1.10.** Características del GPRS del SIM900

Fuente: Data sheet del SIM900

El SIM900 tiene una velocidad máxima de subida de 85.6 kbs, lo que en muchos casos de aplicación M2M o IOT es más que suficiente para satisfacer las necesidades de transmisión o bien de recepción. En la tabla 1.3 se puede apreciar la velocidad para cada esquema sobre el aire.

**Tabla 1.3**

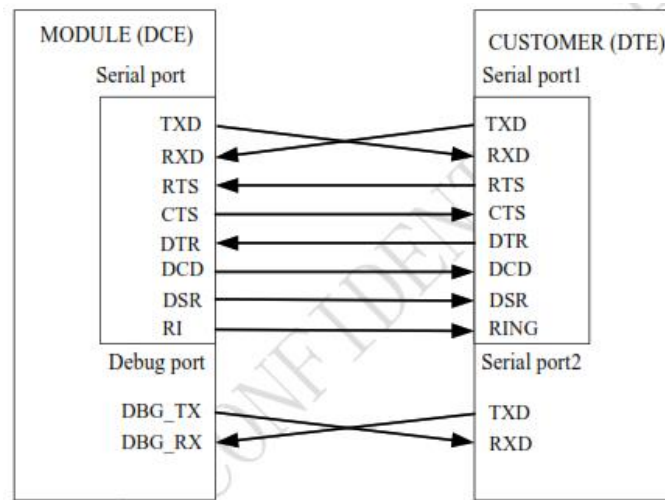
*Códigos de esquema y máximos data rate de la red sobre el aire*

Coding scheme	1 Timeslot	2 Timeslot	4 Timeslot
CS-1:	9.05kbps	18.1kbps	36.2kbps
CS-2:	13.4kbps	26.8kbps	53.6kbps
CS-3:	15.6kbps	31.2kbps	62.4kbps
CS-4:	21.4kbps	42.8kbps	85.6kbps

**Nota.** Fuente: Data sheet del SIM900

El SIM900 tiene un puerto serie UART en los pines<sup>6</sup> 9 (TXD) y 10 (RXD) que se pueden configurar a un baud rate de 1.2 kbps a 115.2 kbps. El módulo GSM está diseñado como un DCE (Data Communication Equipment), siguiendo la tradicional conexión DCE-DTE (Data Terminal Equipment o cliente). El módulo y el cliente (DTE) están conectados a través de las siguientes señales mostradas en la figura 1.11.

<sup>6</sup> Ver pinout en figura A.14 en anexos



**Figura 1.11.** Conexión de las interfaces serie del SIM900

Fuente: Data sheet SIM900

## 1.12. EL MODELO TCP/IP

El modelo TCP/IP es un conjunto de protocolos que permite la comunicación entre dispositivos de la misma red o en redes distintas, independientemente del sistema operativo. Tienen un enfoque de capas las cuales son cuatro, pero se podrían desplegar hasta cinco si incluimos la capa física. (Tanembuam, 2012) En la figura 1.12 podemos observar una ilustración de este modelo.



**Figura 1.12.** Modelo TCP/IP

Fuente: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/modelos/Mtcp.html>

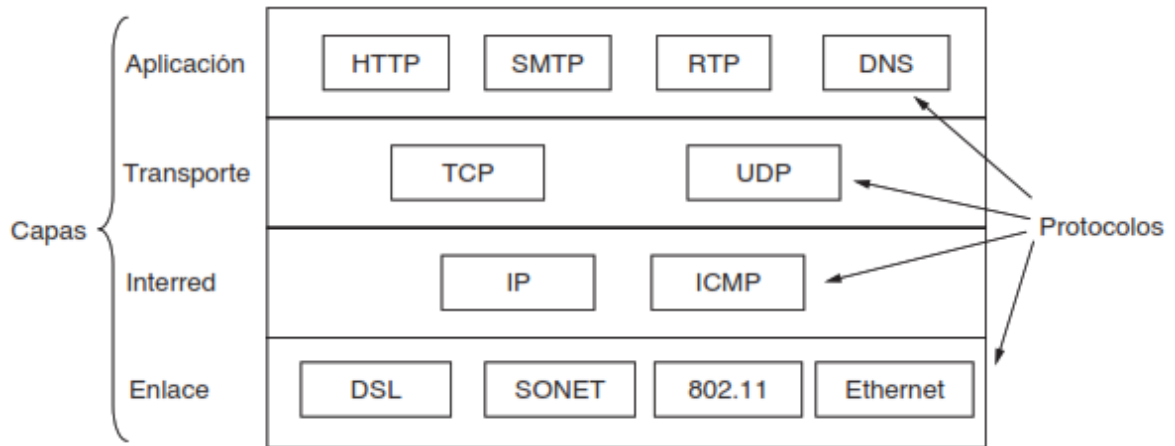
**La capa de enlace o host a red:** esta capa describe que enlaces, como las líneas seriales y Ethernet clásica, se deben llevar a cabo para cumplir con las necesidades de esta capa de interred sin conexión. En realidad no es una capa en el sentido común del término, sino una interfaz entre los hosts y los enlaces de transmisión.

**La capa de interred:** permite que los hosts inyecten paquetes en cualquier red y que viajen de manera independiente hacia el destino (que puede estar en una red distinta). Se debe tener en cuenta que se está usando la palabra interred en un sentido genérico, aunque esta capa esté presente en internet.

**La capa de transporte:** permite que las entidades pares, en los nodos de origen y de destino, lleven a cabo una conversación. Aquí se definieron dos protocolos de transporte de extremo a extremo. El primero, TCP (Protocolo de Control de la Transmisión) y UDP (Protocolo de Datagrama de Usuario).

**La capa de aplicación:** esta contiene todos los protocolos de alto nivel. Aquí se encuentran varios protocolos que los usuarios utilizan con frecuencia, como por ejemplo SMTP, FTP o HTTP (Protocolo de transferencia de Hipertexto). (Tanenbaum, 2012).

En la figura 1.13 se muestran algunos de los protocolos pertenecientes a las capas. En particular haremos hincapié en el protocolo HTTP



**Figura 1.13.** El modelo TCP/IP con algunos de los protocolos

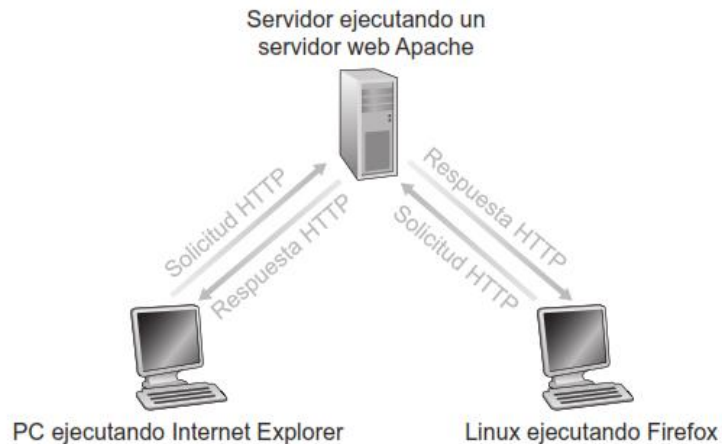
Fuente: Tanenbaum, A. & Wetherall, D. (2012). Redes de computadoras (5ta ed.). Naucalpan de Juárez, México: Pearson

### 1.12.1. EL PROTOCOLO HTTP

El protocolo de transferencia de hipertexto (HTTP), es el protocolo de la capa de aplicación de la Web y se encuentra en el corazón de la Web. HTTP se implementa en dos programas: un programa cliente y un programa servidor. El programa cliente y el programa servidor, que se ejecutan en sistemas terminales diferentes, se comunican entre si intercambiando mensajes HTTP. HTTP define la estructura de estos mensajes y como el cliente y el servidor intercambian los mensajes.

HTTP define como los clientes web solicitan páginas web a los servidores web y como estos servidores transfieren esas páginas web a los clientes. Esta idea se ilustra en la figura 1.14.





**Figura 1.14.** Comportamiento solicitud-respuesta de HTTP

Fuente: Kurose, J. & Ross, K. (2012). Redes de computadoras: un enfoque descendente (5ta ed.). Madrid, España: Pearson.

Cuando un usuario solicita una página web, el navegador envía al servidor mensajes de solicitud HTTP para los objetos contenidos en la página. El servidor recibe las solicitudes y responde con mensajes de respuesta HTTP que contienen los objetos. Para que esto sea posible el cliente HTTP primero debe iniciar una conexión TCP con el servidor. (Kurose, 2010).

Una lista de protocolos es un conjunto de protocolos utilizados por cierto sistema, un protocolo por capa. (Tanenbaum, 2012). El módulo SIM900 cuenta con una lista<sup>7</sup> (stack) TCP/IP embebido que es manejado por comandos<sup>8</sup> AT y habilita el host de aplicación para acceder fácilmente a los servicios de internet HTTP. Esto significa que el SIM900 tiene un conjunto de protocolos embebidos para la comunicación con internet sin importar la plataforma destino.

---

<sup>7</sup> Ver detalles de lista en Anexo B.2

<sup>8</sup> Ver comandos AT en Anexo B.

### **1.13. EL SERVIDOR WEB**

Un servidor Web es un programa diseñado para transferir datos de hipertexto, es decir objetos alojados en las páginas webs. Estos servidores usan el protocolo HTTP.

Los servidores se encuentran montados o alojados en un ordenador con conexión a internet. El servidor se encuentra en todo momento, mientras esté operando, recibiendo peticiones de algún navegador y responde a dichas peticiones con los objetos solicitados. En este sentido podemos hablar de servidores webs de bases de datos, de correo, etc.

Los servidores web procesan archivos escritos en diferentes lenguajes de programación como PHP, Python, Java y otros. (Hostinger)

### **1.14. EL SERVIDOR APACHE**

Apache es un software de servidor web de Apache Software Foundation. Este software es de código abierto, gratuito y multiplataforma que implementa el protocolo HTTP. Su trabajo es establecer una conexión entre un servidor y los navegadores de los visitantes del sitio web (Firefox, Google Chrome, Safari, etc.) mientras envían mucha información entre ellos (estructura cliente-servidor). El servidor y el cliente se comunican a través de protocolo HTTP y Apache es responsable de garantizar una comunicación fluida y segura entre las dos máquinas.

Además de apache existen muchos otros servidores web rivales, pero apache destaca siendo el software que se ejecuta en el 46% de los sitios webs de todo el mundo. (Hostinger, 2019). (Villada, 2014).

Algunas de las ventajas de utilizar Apache son:

- Es de código abierto y gratuito, incluso para uso comercial.
- Software confiable y estable.
- Parches de seguridad regulares y actualizados con frecuencia.
- Flexible debido a su estructura basada en módulos.
- Fácil de configurar.
- Multiplataforma (funciona tanto en servidores Unix como en Windows).

Algunas desventajas pueden ser:

- Problemas de rendimiento en sitios web con demasiado tráfico.
- Demasiadas opciones de configuración pueden generar vulnerabilidades de seguridad. (Hostinger, 2019).

### **1.15. EL HOSTING**

El hosting es un servicio en línea que permite publicar un sitio web, aplicación web o manejar bases de datos remotas. Cuando te registras para un servicio de hosting, básicamente alquilas un espacio en un servidor en el cual puedes almacenar todos los archivos y datos necesarios para que tu sitio web funcione correctamente. (Hostinger, 2019).

### **1.16. PHP**

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de alto nivel y de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

El código se ejecuta en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que

procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber que se tiene debajo de la manga.

El lenguaje PHP es fácil de utilizar para principiantes pero también ofrece características avanzadas para programadores profesionales. Aunque ha sido desarrollado para la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas. (PHP, s.f.).

### **1.17. MYSQL**

MySQL es el sistema de gestión de bases de datos SQL elegida por la gran mayoría de programadores en PHP, Soporta el lenguaje SQL y la conexión de varios usuarios, pero, en general, se utiliza para aplicaciones de tamaño pequeño-medio. (Pavón, 2007).

El software MySQL es un sistema cliente / servidor que ofrece un servidor de bases de datos SQL (lenguaje de consulta estructurado) muy rápido, multiproceso, multiusuario y robusto. MySQL Server puede ejecutarse cómodamente en una computadora de escritorio o portátil, junto con otras aplicaciones, servidores web, etc., que requieren poca o ninguna atención.

El software MySQL tiene doble licencia. Los usuarios pueden elegir usar el software MySQL como un producto de código abierto y gratis o pueden comprar una licencia comercial estándar de Oracle. (MySQL, s.f.).

### **1.18. MIT APP INVENTOR**

App Inventor es una aplicación Web que se ejecuta en navegadores como Google Chrome 4.0 o superior, Apple Safari 5.0 o superior y Mozilla Firefox 3.6 o superior.

App Inventor permite crear aplicaciones fácil y rápido para smartphones y tablets. App inventor fue creado para principiantes que no han programado antes pero que cuentan con buen nivel de razonamiento lógico y matemático.

Con App Inventor se utiliza una pantalla de diseño para visualizar la pantalla de la app, de manera que se utiliza un editor especial conocido como editor de bloques para crear las acciones que la aplicación debe desempeñar. Con el editor de bloques no se tiene que conocer un lenguaje como Java para programar la aplicación. En lugar de eso, se ensamblan visualmente códigos de bloque para crear la app. (Gaddis, 2015).

App Inventor corre en la nube, es decir lo hace en un servidor remoto al cual se accede desde internet. App inventor es parte del **Center for Mobile Learning** del MIT (Massachussets Institute of Technology) así que esta hospedado en servidores que son controlados por el MIT. Los proyectos creados con App Inventor son almacenados en servidores remotos.

### **1.19. ENERGIA SOLAR FOTOVOLTAICA**

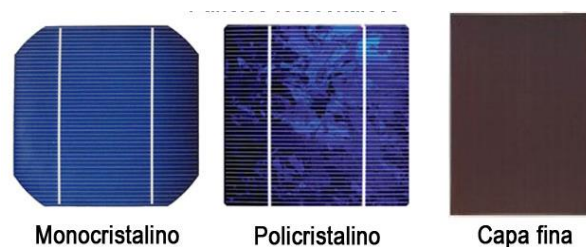
La energía solar fotovoltaica es una tecnología que genera corriente continua (potencia medida en vatios o kilovatios). Por medio de semiconductores cuando estos son iluminados por un haz de fotones. Mientras la luz incide sobre una célula solar, que es el nombre dado al elemento fotovoltaico individual, se genera potencia eléctrica; cuando el haz se extingue, la electricidad desaparece. Las células solares no necesitan ser cargadas como las baterías. (Colegio Oficial de Ingenieros de Telecomunicación, 2002)

### 1.19.1. CELULAS SOLARES

Es el dispositivo en el que se produce la conversión de luz en electricidad gracias a las propiedades de los semiconductores por una parte y a las estructuras que permiten extraer los electrones excitados de la célula, antes de que vuelva a su estado de equilibrio térmico, hacia el circuito exterior para que realicen el trabajo.

#### 1.19.1.1. LAS CELULAS DE SILICIO CRISTALINO

Las células de silicio están hechas precisamente de silicio, el cual es un material muy abundante en la corteza terrestre. Sus características eléctricas lo hacen adecuado para la conversión de luz solar en electricidad. Para ello se requiere que el material sea de gran pureza y perfección estructural, para ello se emplea un material de calidad microelectrónica y obleas monocristalina, policristalinas o capa fina (o amorfo). En la figura 1.15 se muestran estos tipos de celdas.



**Figura 1.15.** Tipos de celdas fotovoltaicas hechas de silicio

Fuente: <http://www.energiasrenovablesinfo.com/solar/tipos-paneles-fotovoltaicos/>

#### 1.19.1.2. INCLINACION Y ORIENTACIÓN DEL MODULO

Para obtener el mayor rendimiento del generador fotovoltaico se ha de procurar que reciba la mayor cantidad posible de luz solar sobre su superficie activa. Y como el sol varía su posición en el cielo cambiando su altura y la inclinación de sus rayos, se debe determinar cuál será la colocación ideal. (Ente Vasco de la Energía, 2000).

En la página de Global Solar Atlas, es posible determinar que la orientación o azimut del módulo solar es de  $180^\circ$  (hacia el sur) con  $15^\circ$  de inclinación.

### **1.19.2. CONTROLADOR DE CARGA Y ACUMULADOR EN SISTEMAS FOTOVOLTAICOS**

En las instalaciones fotovoltaicas autónomas, los módulos solares fotovoltaicos, una vez instalados, siempre se encuentran disponibles para generar electricidad. Sin embargo, la cantidad de radiación solar que reciben se presenta variable, sometida al ciclo diario de los días y las noches, al ciclo anual de las estaciones y a la variación aleatoria del estado de la atmósfera con sus días claros, nubosos, tormentas, etc.

Es por ello que se hace necesario el uso de acumuladores para este fin. El acumulador almacena energía siendo capaz de transformar la energía potencial química en energía eléctrica. En la figura 1.16 se observa un típico acumulador plomo-acido.



**Figura 1.16.** Acumulador plomo-acido de 9AH

Fuente: data sheet del fabricante

De este modo la batería es capaz de suministrar energía cuando el modulo fotovoltaico no está generando suficiente energía dadas las condiciones mencionadas. (Ente Vasco de la energía, 2000).

El controlador de carga por su lado tiene la función de gestionar la corriente eléctrica que absorbe o cede (en CC) el acumulador o batería de acumuladores. Un controlador de carga se observa en la figura 5.30.



**Figura 1.17** Controlador de carga.

Fuente: <https://www.ebay.com/itm/10-20-30A-Dual-USB-Solar-Panel-Battery-Regulator-Charge-Controller-12V-24V-/153000520308>

La función principal del controlador de carga es vigilar el ciclo de carga y descarga del acumulador evitando sobrecargas y sobredescargas así como la carga flotante. (Ente Vasco de la Energía, 2000).



## CAPITULO 2: ANALISIS Y PRESENTACION DE RESULTADOS

### 2.1. ANALISIS DEL SISTEMA

En esta etapa se puso en contexto el problema a través de fuentes que nos permita estudiarlo y caracterizarlo. Esto con el objeto de generar una serie de criterios válidos que nos permita tener una idea de lo que debería realizar el sistema.

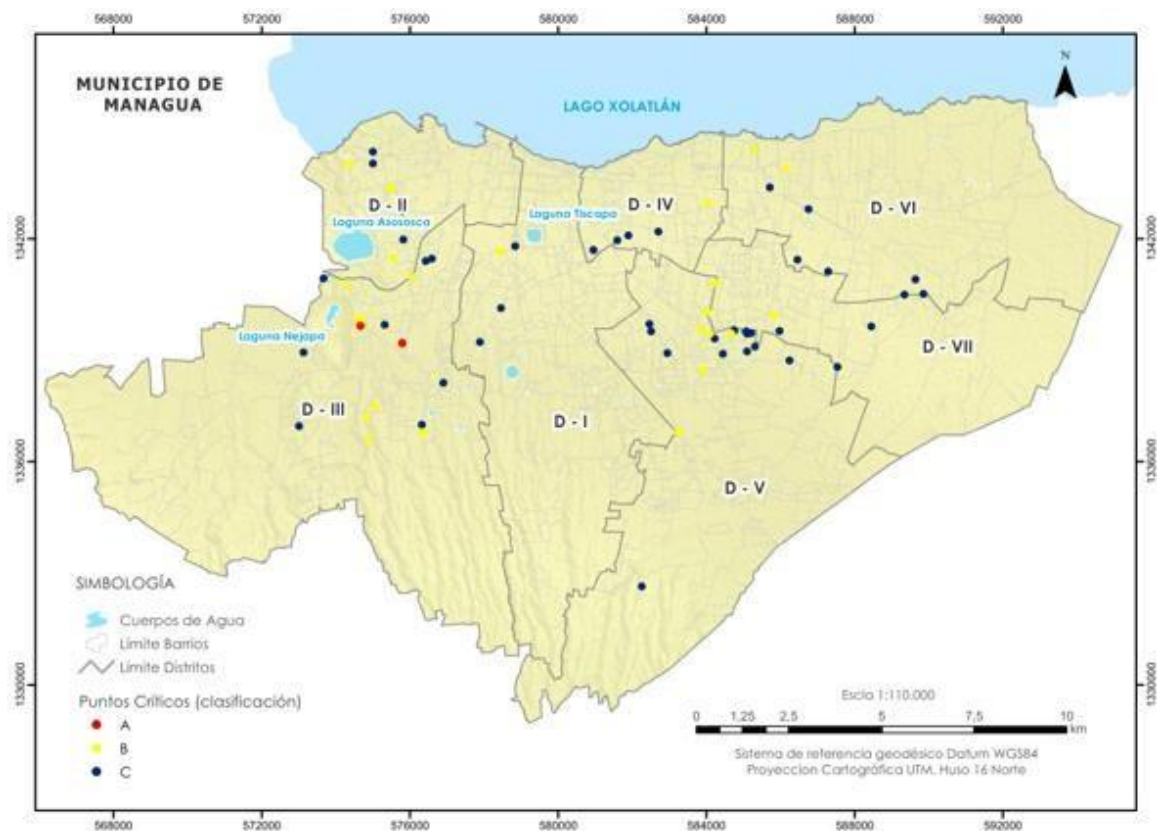
Hay que tomar en cuenta que como puntos críticos, todos guardan casi los mismos riesgos mencionados en el marco teórico, como son deslizamientos de tierra, anegamientos, daños a la infraestructura, etc. Nuestro enfoque estará orientado a las inundaciones pluviales o anegamientos producidos durante fuertes lluvias en la zona en cuestión.

#### 2.1.1. ANALISIS DE FUENTES DE INFORMACION

En la figura 2.1 tomada de la sección de anexos del PMES<sup>9</sup>, **Diagnóstico y Análisis de Puntos Críticos del Municipio de Managua**, se pueden contabilizar unos 64 puntos críticos en total, clasificados de acuerdo tres colores por su nivel de peligrosidad. En el documento se explica que según el nivel de riesgo está clasificado de acuerdo a las letras A, B, C y D o bien a los colores asociados a estas letras, donde la letra A indica el máximo de riesgo de pérdida de vida humana.

---

<sup>9</sup> Plan maestro de escorrentía superficial.



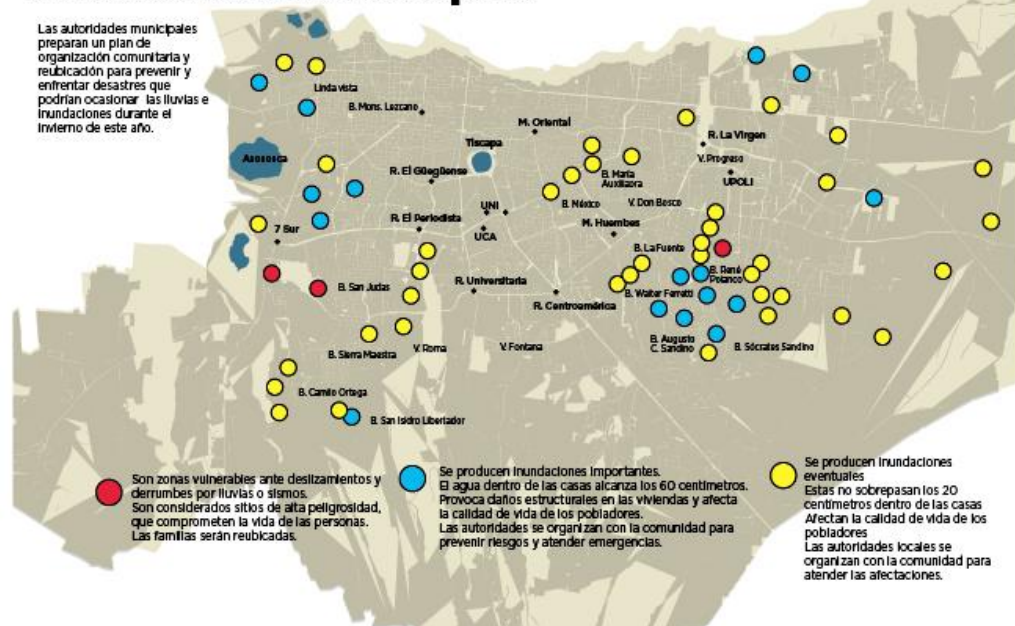
**Figura 2.1.** Ubicación general de los puntos críticos.

Fuente: Amec Foster Wheeler. (2017). PLAN MAESTRO DE ESCORRENTIA SUPERFICIAL (PMES) DEL MUNICIPIO DE MANAGUA. Miami, FI.

Hay que mencionar que la información del PMES estaba restringida al momento de la solicitud, pero dada la valoración que se hizo en ese momento se obtuvo esta importante fuente. Este documento o parte de este no se incluye en los anexos a petición de la Ing. Janixia Palma, encargada de brindar este tipo de información.

En un artículo del Nuevo Diario del 7 de abril de 2018 se tenían contabilizado, según un informe brindado por la alcaldía de Managua, 59 puntos críticos ante lluvias y sismos. El mismo artículo muestra la imagen de la figura 2.2.

## Vulnerabilidad en la capital



**Figura 2.2.** Puntos críticos en Managua

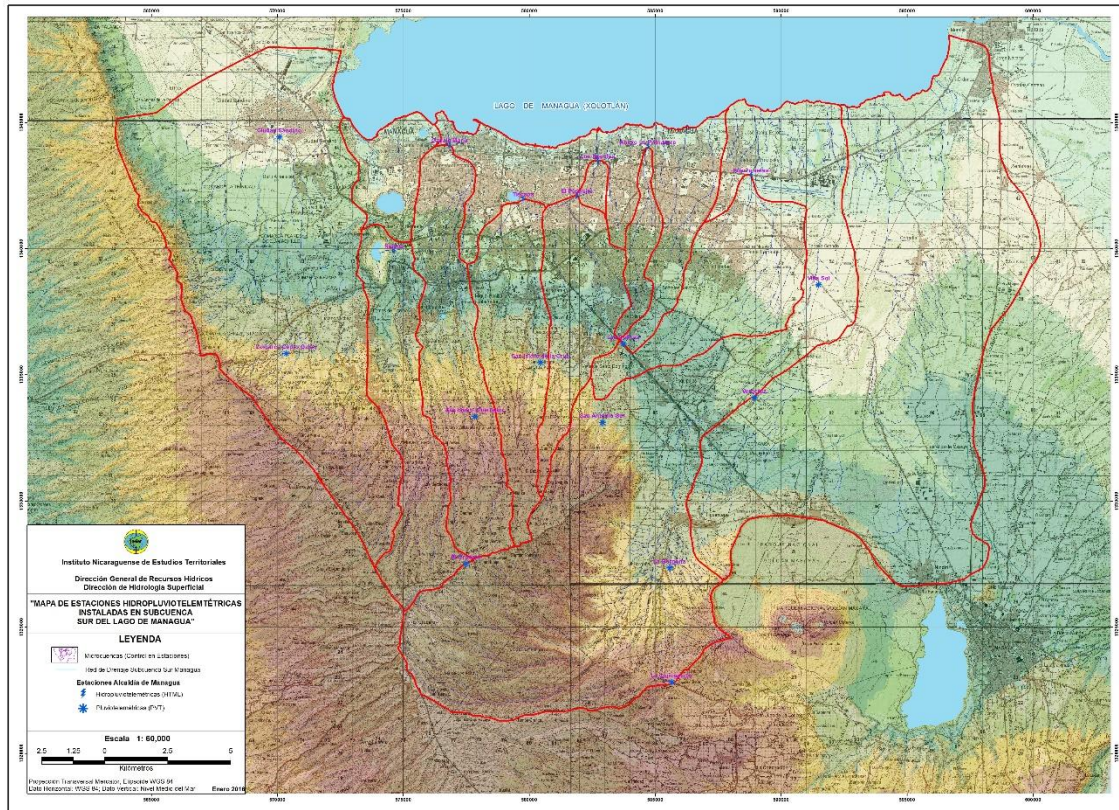
Fuente: García, K. (7 de abril de 2018). Managua tiene 59 puntos críticos ante desastres. El Nuevo Diario. Recuperado de: <https://www.elnuevodiario.com.ni/nacionales/460404-managua-tiene-59-puntos-criticos-desastres/>

Según el artículo, las acciones a tomar, por la alcaldía, para los casos más críticos es el traslado de familias a zonas más seguras. Las familias en riesgo de inundaciones no serán movilizadas, sino que se trabajara en fortalecer la organización comunitaria para prevenir riesgos y atender emergencias.

De la imagen también se puede resaltar que los niveles de agua alcanzados en algunos puntos pudo llegar incluso a los 60 cm.

De lo anterior podemos concluir que aún hay zonas con familias en riesgo ante inundaciones y durante las intensas lluvias su seguridad está en dependencia de que tan rápido puedan responder los sistemas comunitarios organizados para informar sobre una emergencia.

En la figura 2.3, se muestra un mapa de estaciones hidropluviotelemétricas que tienen la función de subir datos de nivel de precipitaciones de agua de lluvia a través de una red telemétrica satelital. En total existen unas 18 estaciones de este tipo ubicadas en los principales cauces de las microcuencas del municipio.



**Figura 2.3.** Mapa de estaciones hidropluviotelemétricas en el municipio de Managua.

Fuente: Alcaldía de Managua.

Según la información suministrada por la alcaldía, las estaciones son propiedad de la alcaldía y los datos obtenidos son requeridos para la planeación y diseño dirigidos a la mejora del sistema de drenaje pluvial del municipio.

Estos datos son los niveles de precipitación de lluvia generados cada 10 o 15 minutos y no pueden ser accedidos gratuitamente por los usuarios, sino que para obtenerlos se debe pagar por ellos.

Por lo tanto no es posible ver datos de precipitaciones ni de nivel de agua en tiempo real gratuitamente en el portal web de INETER y tampoco en ningún otro portal web en la red en Nicaragua.

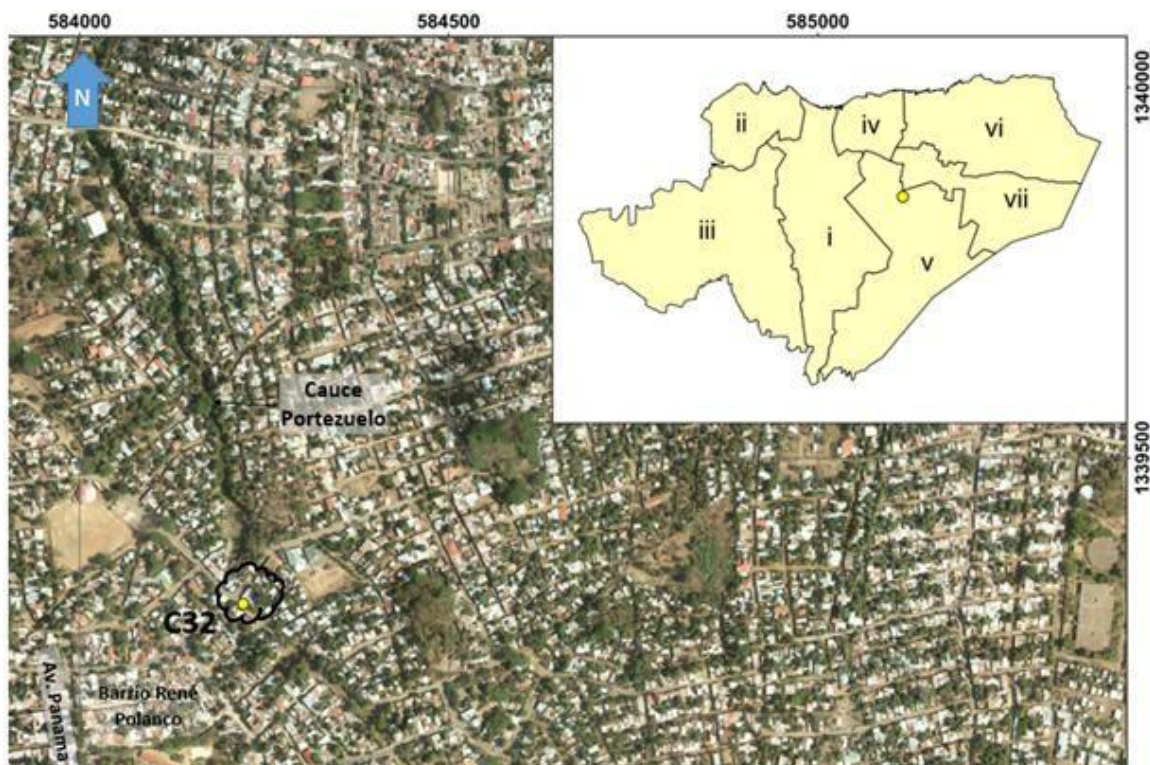
Finalmente se hizo una pequeña encuesta en línea para determinar la importancia o el interés de la sociedad de contar con un sistema en línea y abierto al público para conocer la situación de inundación en alguno de los puntos críticos que tiene el municipio. Por supuesto se hizo notar en la encuesta<sup>10</sup> que se haría uso de una aplicación Android como parte integral del sistema como medio de visualización y monitoreo. El resultado de esta consulta arrojó que el 73% estaba de acuerdo en ello, considerando que sería interesante contar con esta herramienta en línea en el sentido que se puede conocer la situación de nivel de agua durante las inundaciones y que también le puede servir a las autoridades para responder de manera rápida y eficaz en la gestión del riesgo en caso de utilizarla.

Explorando el PMES buscando zonas de anegamiento o inundación estática, para el caso es lo mismo para el objetivo del proyecto, se encontraron varias zonas en esta situación. De forma que seleccionamos cualquiera de ellas, particularmente una zona del distrito V que se puede observar en la figura 2.4.

---

<sup>10</sup> Ver en Anexo G el formato y resultados.





**Figura 2.4.** Barrio anexo Jardines de Veracruz. Punto vulnerable C32

Fuente: Amec Foster Wheeler. (2017). PLAN MAESTRO DE ESCORRENTIA SUPERFICIAL (PMES) DEL MUNICIPIO DE MANAGUA. Alcaldía de Managua.

Esta zona afectada o punto crítico está marcada con un código C32 (según el PMES). Este lugar está ubicado en el barrio Anexo Jardines de Veracruz, junto al complejo Concepción Palacios. En esta zona la escorrentía busca naturalmente el cauce, sin embargo se ve bloqueada por el asentamiento. Existen 34 familias afectadas la cuales se ven frecuentemente anegadas ante las intensas lluvias.

### 2.1.2. CRITERIOS DE DISEÑO

Teniendo lo anterior en cuenta se determinó que en general el sistema debe hacer lo siguiente:

- Medir el nivel de agua en la zona indicada.
- Implementar un servidor en línea.

- Subir datos de nivel al servidor.
- Implementar una base de datos.
- Recuperar desde el servidor el valor de nivel de agua.
- Implementar una aplicación Android con interfaz amigable.
- Visualizar estos datos de nivel en tiempo real.
- Emitir alerta de acuerdo a ciertos criterios.
- Implementar un sistema con alimentación autónoma (opcional).

## **2.2. DISEÑO DEL SISTEMA**

Esta sección comprende el proceso de diseño del sistema, teniendo en cuenta el hardware, software, simulación, técnicas de diseño, elaboración de tarjeta y procesos de configuración del hardware y la programación.

Antes de entrar en el proceso en si del diseño del sistema, con sus diagramas de bloque, flujogramas y otros diagramas asociados, se hará una selección de componentes de hardware y software que formarán parte del sistema. La selección se hará de acuerdo a los criterios diseño del sistema y la factibilidad técnica y económica. Esto permitirá tener elementos de juicio al momento de la selección.

### **2.2.1. ESTUDIO DE FACTIBILIDAD DE LOS PRINCIPALES COMPONENTES.**



Para determinar la factibilidad técnica y económica de los componentes de sistema debemos tomar en cuenta los criterios de diseño del sistema que nos permitirá determinar los componentes más adecuados para este sistema y validar varios de los principales componentes sometidos a elección. Cada tipo de componente será comparado con uno o dos componentes similares para determinar cuál es el que presenta las mejores características respecto a los criterios de diseño como del factor técnico y costo.

### 2.2.1.1. SELECCION DEL SENSOR DE NIVEL

La siguiente tabla muestra las características de los sensores que se utilizan para medir nivel de líquidos.

**Tabla 2.1**

*Comparativa de los sensores de nivel*

Sensor	Descripción	Características generales	Precio
 MPX4115AP	Sensor integrado de presión para aplicaciones de altímetro y barómetro (También usado para medir nivel)	<ul style="list-style-type: none"> <li>Rango 15 a 115 kPa (2.18 a 16.7 psi)</li> <li>Medida absoluta</li> <li>Especialmente diseñado para usar con <math>\mu</math>controladores y <math>\mu</math>procesadores</li> <li>Voltaje de alimentación: 4.85 V a 5.35 V (típico: 5 V)</li> <li>Corriente típica: 7 mA</li> <li>Sensibilidad: 46 mV/kPa.</li> <li>Precisión de <math>\pm 1.5</math> kPa</li> </ul>	C\$ 37 + envío (aprox. C\$ 10)
 HC-SR04*	Sensor de ultrasonidos de no contacto utilizado para medir distancias.	<ul style="list-style-type: none"> <li>Mide en un rango de 2 cm-400 cm</li> <li>Precisión de hasta 3mm.</li> <li>DC 5V</li> <li>Corriente: 15 mA.</li> </ul>	C\$ 2.25 + envío (C\$ 10) Local: C\$ 5

**Nota.** Aquí se encuentra información obtenida de los data sheets y cotización de precios.

\* Ver características más detalladas en data sheet en Anexos. Fuente: autor.

Como se puede notar de la tabla 2.1, la mejor opción a tomar es el sensor de ultrasonidos debido al bajo precio y a la precisión de la medida. Respecto a su precisión tiene un error máximo de 3mm, pero esto es mejor considerando que la precisión del sensor de presión puede ser de hasta  $\pm 10\%$ , eso sin considerar el efecto de la temperatura sobre la presión de una ampolla que se encuentre conectada al sensor de presión. A mayor temperatura mayor presión en esta. A pesar del menor consumo de corriente respecto al sensor ultrasónico, eso no es algo muy incidente en la selección.



### 2.2.1.2. SELECCION DEL MICROCONTROLADOR

Dado el mayor uso de microcontroladores PIC así como la variedad de estos y el manejo sencillo de compiladores en lenguaje c de alto nivel es viable el uso de microcontroladores PIC, además que se cuenta con abundantes recursos bibliográficos para su configuración.

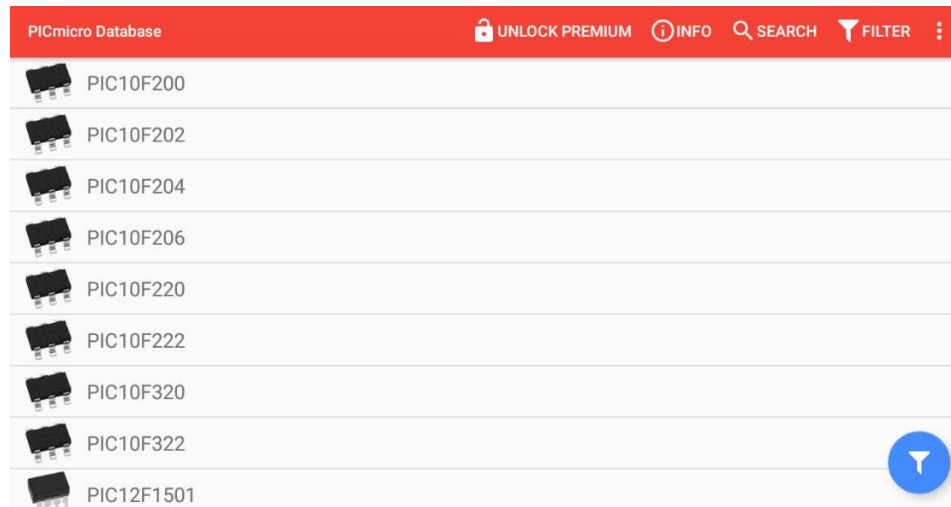
ElectroDroid es una aplicación Android para sistemas operativos Android e iOS que nos permite filtrar las características técnicas deseadas en la búsqueda del microcontrolador PIC. Para este caso usaremos su versión limitada.



**Figura 2.5.** Interfaz de inicio de la aplicación ElectroDroid

Fuente: Autor

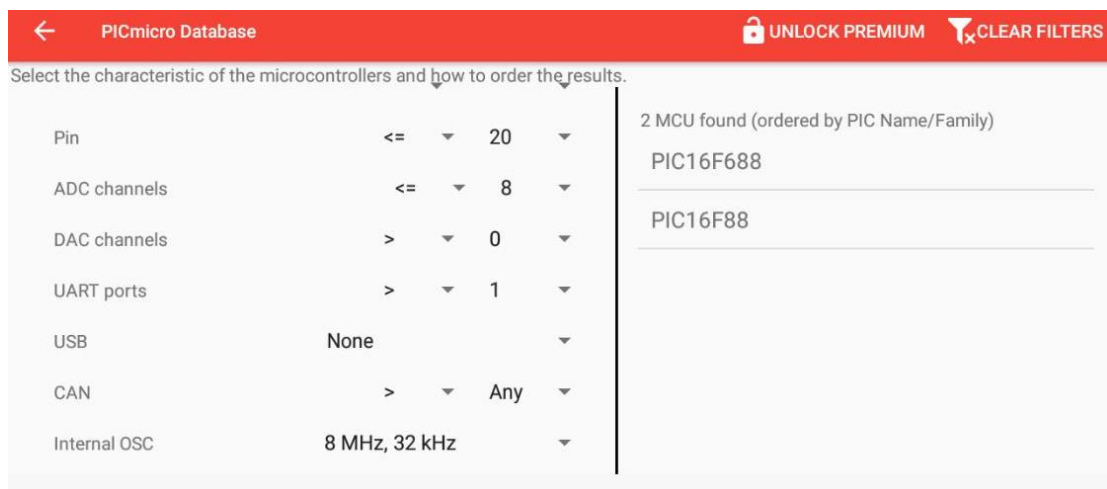
Ingresando en la pestaña “RECURSOS” y luego en la opción “Base de Datos de Microcontroladores PIC”, nos lleva a la pantalla que se muestra en la figura 2.6.



**Figura 2.6.** Base de datos de los microcontroladores PIC.

Fuente: Autor.

En la pantalla de la figura 2.6 ingresamos en la pestaña “FILTER” y llenamos los campos con las características requeridas, como se muestra en la figura 2.7.



**Figura 2.7.** Pantalla de resultados de búsqueda.

Fuente: Autor.

Como se aprecia en la figura 2.7, al lado izquierdo, algunas de los parámetros técnicos han sido filtradas y en el lado derecho aparecen todos los posibles resultados de los PICs que cumplen con estos parámetros.




El resultado son dos PICs, de los cuales se ha elegido el PIC16F688 por razón de su reducido tamaño, pues tiene 14 pines, dos menos que el PIC16F88, y su precio es más reducido, \$5 del PIC16F688 contra \$6.31 del otro.

### 2.2.1.3. SELECCION DEL MODULO DE TRANSMISION

En la tabla 2.2 se muestran los módulos de transmisión-recepción a tomar en cuenta en la selección.

**Tabla 2.2**

*Comparativa de los módulos de transmisión*

Modulo	Descripción	Características	Precio
 <p>SIM900*</p>	<p>Módulo GSM/GPRS inalámbrico SIM900 cuatro bandas 850/900/1800/1900 MHz</p>	<ul style="list-style-type: none"> <li>• Cobertura: Definido por la cobertura de la red telefónica.</li> <li>• Voltaje: 5 V.</li> <li>• Corriente: 2 A (Pico en tx). 360 mA (Típico).</li> <li>• Interface: UART (CMOS).</li> <li>• Banda GSM/GPRS: 850/900/1800/1900 MHz.</li> <li>• Máxima velocidad de transmisión de datos: 85 Kbps (subida) y 42 Kbps (bajada).</li> <li>• Comunicación con comandos AT.</li> </ul>	<ul style="list-style-type: none"> <li>• \$24.51</li> <li>• Con gastos de envío (\$34.51)</li> </ul>
 <p>X-bee pro (dos módulos)</p>	<p>Módulo de RF que implementa el protocolo 802.15.4 y que opera en 2.4 GHz.</p>	<ul style="list-style-type: none"> <li>• Cobertura: 1600 m (línea de vista) y 750 m (con obstáculo).</li> <li>• Voltaje de operacion: 3.3 V</li> <li>• Corriente: 250-340 mA (Tx pico). 45 mA (Tx típico). 55 mA (Rx).</li> <li>• Velocidad de transmisión: 250 Kbps.</li> <li>• Frecuencia: 2.4 GHz.</li> <li>• Soporta varios software como XCTU, plataforma XBee, Waspote y arduino.</li> </ul>	<ul style="list-style-type: none"> <li>• \$43 x 2</li> <li>• Con gastos de envío (\$52.21 x 2)</li> </ul>
 <p>ESP8266</p>	<p>El NodeMCU es una plataforma abierta IoT. Incluye el firmware que corre en el chip ESP8266 Wi-Fi. Trabaja muy bien con Arduino IDE/Micropython.</p>	<ul style="list-style-type: none"> <li>• Cobertura: depende de la intensidad de la señal Wi-Fi.</li> <li>• Corriente: 12-200 mA.</li> <li>• Un PWM, I2C, SPI y UART.</li> <li>• CP2102 usb integrado.</li> <li>• Usa el protocolo inalámbrico 802.11b/g/n.</li> <li>• Soporta los lenguajes Lua y Arduino C y comandos AT.</li> </ul>	<ul style="list-style-type: none"> <li>• \$8.39</li> <li>• Con gastos de envío: (\$18.39)</li> </ul>

- 
- Frecuencia 2.5 GHz.
  - Máxima velocidad de transmisión: 54Mbps (Tx) 54 Mbps (Rx).
- 

**Nota.** Aquí se encuentra información obtenida de los data sheets y cotización de precios

Fuente: Autor.

Se ha seleccionado el módulo GSM/GPRS dado su reducido tamaño, bajo precio, buena cobertura y es capaz de subir datos directamente a un servidor. Esto último representa una gran ventaja respecto a los otros módulos. Tanto el ESP8266 y el Xbee-pro consumen menos corriente y son de tamaño reducido sin embargo presentan desventajas notables. El módulo ESP8266 requiere tener acceso a una red Wi-Fi con acceso a internet, esto lo hace dependiente de una red o router cercano que siempre se encuentre activo. El módulo Xbee-pro además de requerir dos módulos para una conexión punto a punto para transmitir y recibir datos, no sube los datos directamente al servidor en línea, además que al utilizar dos de estos módulos eleva mucho los costos del sistema.

La principal desventaja del módulo GSM/GPRS es que consume mayor corriente que los demás módulos. Puede llegar a consumir hasta 2A pico en transmisión pero en intervalos muy cortos de tiempo. Sin embargo esta desventaja es salvable respecto a las ventajas ya mencionadas.





El módulo GSM/GPRS tiene un rango que depende de la cobertura de la telefonía celular local. En este caso todas las redes telefónicas tienen cobertura en la mayor parte de municipio de Managua, incluyendo el sitio en el que se pretende ubicar el sistema. Además de eso no se requiere hardware adicional como antena y acondicionador de señal ni de hardware para subir datos al servidor. Este módulo tiene un stack TCP/UDP integrado en su firmware que se encarga de la comunicación directa con el servidor.

### 2.2.1.3.1. SELECCIÓN ENTRE DIVERSOS MODULOS GSM/GPRS

En el mercado existen diversos módulos GSM/GPRS la mayoría fabricados por SIMCOM aunque existen otros con características similares como los fabricados por Ai Thinker. Es por esto que haremos una nueva tabla (tabla 2.3) para decidirnos por alguno de ellos, aquel que mejor se adapte a los requerimientos del sistema. Los parámetros a tomar en cuenta es que debe ser ligero, pequeño, de bajo costo, factible técnicamente y también considerar el tiempo de entrega.

**Tabla 2.3**

*Comparativa entre diversos módulos GSM/GPRS*

Módulo	Precio	Características
Shield SIM900 para arduino uno. 	\$ 86 (Importado)	<ul style="list-style-type: none"> <li>• Entrada para auricular</li> <li>• Entrada para micrófono</li> <li>• Shield para arduino</li> <li>• Red 2G</li> <li>• Cuatro bandas</li> <li>• Comandos AT</li> <li>• Stack TCP/IP en HTTP</li> </ul>
Módulo GSM/GPRS SIM900 	\$ 43 (Importado)	<ul style="list-style-type: none"> <li>• Tamaño pequeño</li> <li>• Liviano</li> <li>• Red 2G</li> <li>• Cuatro bandas</li> <li>• Comandos AT</li> <li>• Stack TCP/IP en HTTP</li> </ul>
Módulo GSM/GPRS A6 	\$ 23 (Importado)	<ul style="list-style-type: none"> <li>• El más pequeño y liviano</li> <li>• Red 2G</li> <li>• Cuatro bandas</li> <li>• Comandos AT</li> <li>• Stack TCP/IP</li> </ul>
Shield SIM900 para arduino uno. 	\$ 32 (Local) \$ 34 (Importado)	<ul style="list-style-type: none"> <li>• Entrada para auricular</li> <li>• Entrada para micrófono</li> <li>• Shield para arduino</li> <li>• Red 2G</li> <li>• Cuatro bandas</li> <li>• Comandos AT</li> <li>• Stack TCP/IP en HTTP</li> </ul>

**Nota.** Aquí se encuentra información obtenida de los data sheets y cotización de precios

Fuente: Autor.

Como se observa en la tabla 2.3 desde el punto vista técnico el SIM900 de C\$ 43 presenta las mejores características ya que no tiene hardware adicional integrado, como el resto de los SIM900, mismos que no son necesarios para la transmisión serie. Además de eso es uno de los más ligeros en peso y es energéticamente más eficiente pues no tiene regulador de tensión integrado. Sin embargo tiene un precio superior a los demás SIM900, además de la incertidumbre sobre su tiempo de entrega.

El módulo A6 aunque es el más barato de todos no cuenta con una robusta serie de comandos integrados en su firmware que permita una sencilla y eficiente conexión con el servidor.

Por lo anterior explicado se ha elegido el shield SIM900 que tiene un precio de C\$ 32, el más bajo de un proveedor local, cuenta con el repertorio de comandos AT requeridos y el soporte HTTP para manejar la comunicación con el servidor web de forma sencilla y eficiente.

#### 2.2.1.4. SELECCION DEL SISTEMA DE ALIMENTACIÓN

En la tabla 2.4 se muestra una comparativa entre un sistema de alimentación fotovoltaico frente a una alimentación proveniente de la red eléctrica.

**Tabla 2.4**

*Comparativa entre sistemas de alimentación*

Sistema	Ventajas	Desventajas
Fotovoltaico	<ul style="list-style-type: none"> <li>Alimentación autónoma.</li> <li>Uso de energía limpia.</li> </ul>	<ul style="list-style-type: none"> <li>Inversión inicial elevada.</li> <li>El sistema es más grande dado el volumen ocupado por el modulo solar y la batería.</li> </ul>
Red eléctrica comercial	Bajo costo a corto plazo.	<ul style="list-style-type: none"> <li>Tiene un costo de instalación por parte del proveedor de servicio eléctrico.</li> <li>El sistema requiere circuitería extra para reducir y rectificar la corriente proveniente de la red y circuito de protección al operar con voltaje a 120 V de alterna.</li> </ul>

**Nota.** Fuente: autor

Como se observa de la tabla 2.4 tiene sus ventajas alimentar con sistemas fotovoltaicos. Iniciando porque no se tendrá que manipular voltajes de 120V incurriendo en un costo por la instalación de los cables de tensión alterna y el medidor del mismo. Además que se alimenta el circuito con energía proveniente del sol, lo cual beneficia el medio ambiente ya que se emplea energía limpia.

### 2.2.1.5. SELECCION DEL SOFTWARE

La factibilidad del software para las aplicaciones que se usarán para este proyecto gira en torno a:


- Aplicaciones de fácil uso dada su sencillez de manejo.
- Uso de versión demo, gratuita o software libre y código abierto.
- Conocimiento medio en lenguajes de programación y uso de bases de datos.

#### 2.2.1.5.1. SELECCION DEL COMPILADOR PARA EL PIC

En la tabla 2.5 se muestran las ventajas del compilador CCS C para escribir y compilar el programa del PIC.

**Tabla 2.5**

*Ventajas del compilador CCS C*

	Principales ventajas respecto a otros compiladores
Compilador CCS C  	<ul style="list-style-type: none"> <li>• Genera un código compacto y eficiente.</li> <li>• Incorpora las librerías necesarias para escribir fácilmente código en C para el PIC.</li> <li>• Tiene versión de prueba.</li> <li>• No usa lenguaje ensamblador, lo que reduciría las líneas de código y la complejidad del programa.</li> <li>• Se integra con otros programas como PROTEUS para el proceso de depuración.</li> </ul>

Nota. Fuente: adaptado de

García, E. (2008). *Compilador C CCS y simulador PROTEUS para Microcontroladores PIC* (1ra ed.). México D.F, México: Alfaomega.

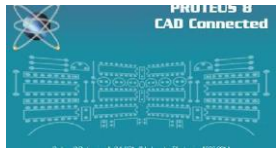
### 2.2.1.5.2. SELECCION DEL SIMULADOR DE CIRCUITOS

Proteus 8.1 es un simulador de circuitos electrónicos con una interfaz muy amigable y fácil de utilizar. En la tabla 2.6 se muestran algunas de sus mejores características.

**Tabla 2.6**

*Ventajas del simulador proteus.*

Principales ventajas respecto a otros simuladores	
Simulador PROTEUS	<ul style="list-style-type: none"><li>• Entorno de diseño gráfico de esquemas electrónicos (ISIS) fácil de utilizar</li><li>• Entorno de diseño de placas de circuito impreso (ARES).</li><li>• Tiene versión de prueba.</li><li>• Es capaz de simular muchos IC entre ellos los microcontroladores de Microchip</li></ul>



Nota. Fuente: adaptado de


García, E. (2008). *Compilador C CCS y simulador PROTEUS para Microcontroladores PIC* (1ra ed.). México D.F, México: Alfaomega.

### 2.2.1.5.3. SELECCION DE LA HERRAMIENTA DE DESARROLLO DE APPS

En la siguiente tabla de tienen las principales ventajas del uso este entorno de desarrollo de aplicaciones Android.

**Tabla 2.7**

*Ventajas de MIT App Inventor*

Principales ventajas	
	<ul style="list-style-type: none"><li>• Interfaz de diseño fácil de utilizar.</li><li>• Editor de bloques para programar las Apps por lo que no se necesita manejar el lenguaje java.</li><li>• Software libre.</li></ul>

**Nota.** Fuente: adaptado de

Gaddis, T. & Halsey, R. (2015). *Starting out with App Inventor for Android* (1ra ed.). Harlow, England: Pearson.




El uso de una herramienta de desarrollo de aplicaciones Android no necesariamente requiere tener amplios conocimientos en materia de programación, siempre y cuando ésta pueda cumplir con el objetivo de crear una App que tenga las características planteadas en los criterios de diseño. Este rol lo cumple muy bien MIT App Inventor.

#### 2.2.1.5.4. SELECCION DEL LENGUAJE DE PROGRAMACION WEB

La tabla 2.8 muestra las principales ventajas del lenguaje PHP respecto a los demás lenguajes de programación. Una de los puntos fuertes del uso de este programa es que es ampliamente utilizado para la programación web, fácil de usar y además es de software libre. Esto lo hace de uso práctico para las necesidades de nuestro sistema.

**Tabla 2.8**

*Principales ventajas de PHP*

	Principales ventajas
	<ul style="list-style-type: none"> <li>• Software libre y open source.</li> <li>• Multiplataforma</li> <li>• Fácil aprendizaje</li> <li>• Posee gran demanda laboral</li> <li>• Amplia documentación en su página oficial</li> </ul>


**Nota.** Fuente: <https://www.jlaja.com/caracteristicas-y-ventajas-del-lenguaje-php/>

#### 2.2.1.5.5. SELECCION DEL GESTIONADOR DE BASES DE DATOS

Para gestionar bases de datos se requiere un administrador de bases de datos, en este caso MySQL o bien utilizar una herramienta desarrollada en PHP para gestionar y administrar las bases de datos MySQL como es PHPMyAdmin.

En nuestro caso se ha optado por usar PHPMyAdmin para gestionar nuestra base de datos dadas las ventajas mostradas en la tabla 2.9.

**Tabla 2.9***Ventajas de phpMyAdmin para gestión de BD*

	Principales ventajas
	• Interfaz web intuitiva
	• Soporte para la mayoría de las características de MySQL: Explorar, eliminar, renombrar y modificar bases de datos.
	• Software libre.
	• Importar datos desde archivos CSV y SQL.
	• Exportar datos a varios formatos: CSV, SQL, XML, PDF.
	• Crear peticiones complejas usando Query-by-example (QBE)
	• Búsquedas globales en una base de datos o subconjunto de la misma.

**Nota.** Fuente: <https://mape309site.wordpress.com/2017/10/24/ventajas-y-desventajas-de-mysql-phpmyadmin-visual-foxpro-y-microsoft-access/>

### 2.2.1.6. SELECCIÓN DEL TIPO DE SERVIDOR

En este apartado se determina el tipo de servidor a utilizar, es decir si este servidor debería ser montado en nuestra pc o bien utilizar un servidor proporcionado por un web hosting. La tabla 2.10 muestra una comparativa al usar los dos tipos de servidores

**Tabla 2.10***Tabla comparativa de tipos de servidores*

Característica	Tipo de instalación del servidor	
	En nuestra PC	En un web hosting* (de pago)
Tiempo de operación diario	24 horas	24 horas
Ancho de banda	Limitado	Ilimitado
Requerimiento de router	Necesario (costo implicado)	No necesario
Requerimiento de dominio	Necesario (costo implicado)	Necesario (costo implicado)
Uso de datos de internet	Necesario	No necesario
Probabilidad de caída del servidor por fallo energético	Alta	Baja
Consumo energético (cualificación)	Alto	Inexistente
Costo de mantener activo el Servidor (cualificación)	Inexistente	Alto

**Nota.** \*El web hosting tomado en cuenta es Hostinger.

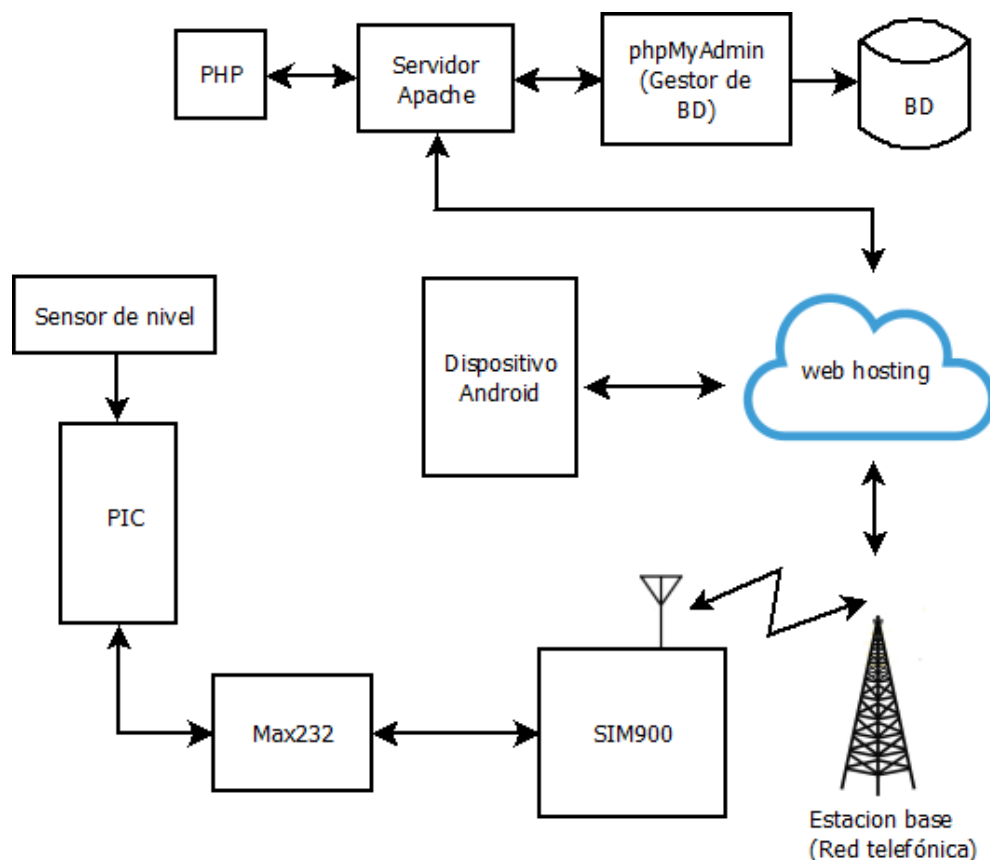
La diferencia en costo total de usar uno y otro tipo de servidor puede llegar a ser grande a favor del servidor con web hosting, principalmente por el consumo energético del PC que aumenta el costo la factura de electricidad. En general también es más seguro usar un hosting ya que hay menos probabilidad de que se caiga el servidor ante un fallo en el sistema. Luego al usar un servidor instalado en PC se debe considerar que este no puede ser montado apropiadamente en cualquier servidor sin el uso de un router, lo que eleva el costo del servidor. Si se usa una Raspberry Pi para montar nuestro servidor, se debe tener en cuenta nuevamente el costo de este, el del router y de los datos de internet, a menos que el sistema opere por mucho tiempo (meses incluso años.).

Aunque el uso de un web hosting parece ser la mejor opción a corto plazo, considerando que los objetivos del proyecto no restringen el tipo de servidor a usar y tomando en cuenta el presupuesto limitado para el sistema en general se ha optado por utilizar una variación del web hosting el cual es un web hosting gratis, específicamente “000webhost” impulsado precisamente por Hostinger. Las características claves de este web hosting son:

- No tiene costo de uso.
- Tiempo de operación: 23 h/día
- Ancho de banda limitado.
- Soporte para PHP; phpMyAdmin y Apache.

### **2.2.2 DESCRIPCION DEL SISTEMA**

Globalmente el sistema se presenta según el diagrama modular de la figura 2.8. En el diagrama cada bloque no representa necesariamente un módulo de hardware sino que puede ser un módulo de software de forma que se puedan establecer las relaciones entre todos los componentes.



**Figura 2.8.** Diagrama de bloques del sistema.

Fuente: autor

Todo inicia en el PIC, previamente programado por el usuario, donde este realiza una lectura de nivel de líquido. Este dato en formato entero es enviado por medio del microcontrolador, usando su puerto serie, hacia el módulo GSM/GPRS, no sin antes hacerlo pasar por el IC MAX232 con el objetivo de garantizar que no exista incompatibilidad de niveles de tensión durante el proceso de comunicación entre ellos. El modulo inalámbrico transfiere el dato hacia la nube aprovechando su lista o pila TCP/IP en servicio HTTP. El módulo SIM900 tiene un socket para insertar una SIM de una operadora local, a la cual se debe activar el servicio de internet lo cual tiene un costo monetario implicado. Una vez en el servidor web, montado en nuestro web hosting, un script en PHP programado por el usuario se encarga de grabar este dato junto con la fecha y hora en que fue subido al servidor. Después con cualquier dispositivo Android con acceso a internet y que tenga instalada la aplicación asociada se podrá visualizar el contenido del dato en tiempo real.

La aplicación no solo debe ser capaz de mostrar el dato de nivel de líquido en tiempo real sino que también mostrará la fecha y hora en que se subió el dato para tener una referencia de tiempo. Además se podrá visualizar la razón de subida del nivel de agua y hacer consultas a las bases de datos especificando cierto criterio de búsqueda. También tendrá un cuadro indicador de alertas teniendo en cuenta un criterio de alertas.

### 2.2.3 DISEÑO DEL SISTEMA DAQ

El sistema debe realizar periódicamente la lectura de una variable de medición, el nivel de agua por medio del sensor de ultrasonidos HC-SR04. Aunque este sensor es usado para medir distancias, el dato generado por este será procesado de manera que pueda medir nivel de agua.

La ecuación<sup>11</sup> para medir distancias con este módulo es la siguiente:

$$Test\ distance = \frac{\left( high\ level\ time \times velocity\ of\ sound \left( \frac{340M}{S} \right) \right)}{2} \quad (Ec. 2.1)$$

O bien usando la siguiente ecuación modificada, para el cálculo en milímetros

$$Test\ distance = \frac{High\ level\ time\ (uS)}{5.8} \quad mm \quad (Ec. 2.2)$$

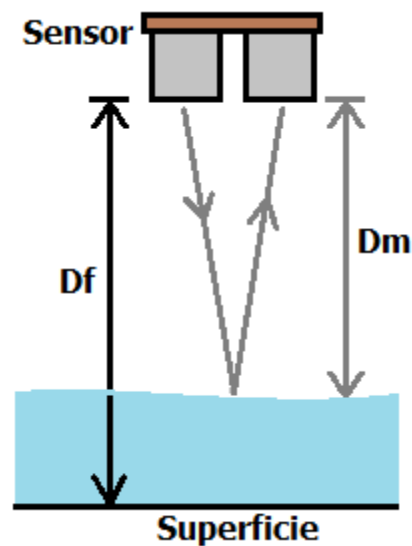
Se debe enviar un pulso de al menos 10uS en el pin Trigger del sensor y este responderá con un pulso en el pin Echo. El tiempo que dure el pulso será el tiempo medido desde que el sensor genera un sonido ultrasónico dirigido hacia el objetivo y rebota hasta que regresa al sensor. Este tiempo de estado en alto en el pin Echo debe ser medido para mayor exactitud en microsegundos, por lo tanto el

---

<sup>11</sup> Ver ecuación en Anexo A.1.

microcontrolador responsable de la medición de este tiempo debe medirlo también en microsegundos.

Dado que el sensor presenta variaciones en su medición, es decir no siempre proporciona la misma distancia en distintas mediciones, se debe definir una rutina de calibración para definir una distancia fija inicial a la cual ha sido colocada el sensor del objetivo. Esto con el propósito de definir una distancia de referencia desde la cual se hagan las mediciones del nivel de agua. Dicha rutina debe ser definida en el microcontrolador.



**Figura 2.9.** Distancia de referencia  $D_f$  y distancia medida  $D_m$ .

Fuente: autor

En la figura 2.9 se muestra la distancia de referencia ( $D_f$ ) y la distancia medida por el sensor ( $D_m$ ). En nivel de agua estará dado por la diferencia entre  $D_f$  y  $D_m$ , en ecuación sería:

$$\text{Nivel} = D_f - D_m \quad (\text{Ec. 2.3})$$

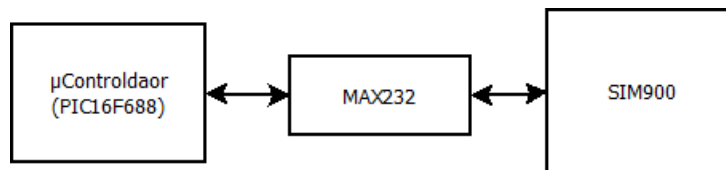
Una vez obtenida esta diferencia, en microsegundos o procesada aritméticamente en mm en el PIC, será enviada como un dato entero a través del sistema hacia el servidor.

## 2.2.4 DISEÑO DEL SISTEMA DE COMUNICACIÓN Y TRANSMISION

Dado que se ha optado por utilizar el módulo GSM/GPRS SIM900 cuyo hardware es capaz de realizar la función de comunicación y transmisión en un solo módulo embebido, se tratarán ambos sistemas de comunicación y transmisión a la misma vez. Ambos sistemas se explican por partes en los siguientes subcapítulos.

### 2.2.4.1 SISTEMA DE COMUNICACION

El sistema de comunicación comprende la comunicación serie entre el microcontrolador y el módulo GSM/GPRS tomando en cuenta el hardware requerido para esta labor y los comandos implicados.



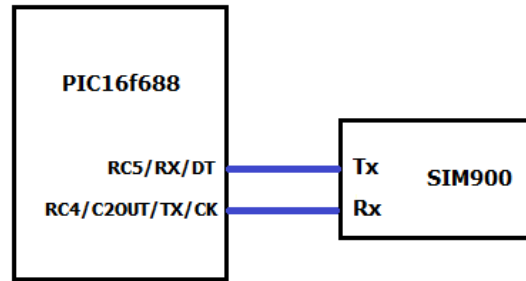
**Figura 2.10.** Sistema de comunicación.

Fuente: autor

Desde el punto de vista del hardware el PIC16F688 tiene un bus serie USART que le permite la transmisión serie de datos. La velocidad de transmisión de datos esta dado en baudios (bits/s) y depende de la frecuencia del oscilador interno o del oscilador externo. El baud rate para el PIC16F688 está definido por la siguiente ecuación tomada del datasheet:

$$\text{Desired Baud Rate} = \frac{F_{osc}}{64([SPBRGH:SPBRG] + 1)} \quad (\text{Ec. 2.4})$$

Donde SPBRGH:SPBRG es un registro par que se debe configurar con un valor entero para obtener el baud rate deseado. Pero claro la configuración de estos registros se realiza automáticamente al usar directivas del compilador CCS C. La forma básica de interconexión con el módulo GSM/GPRS se muestra en la figura 2.11.

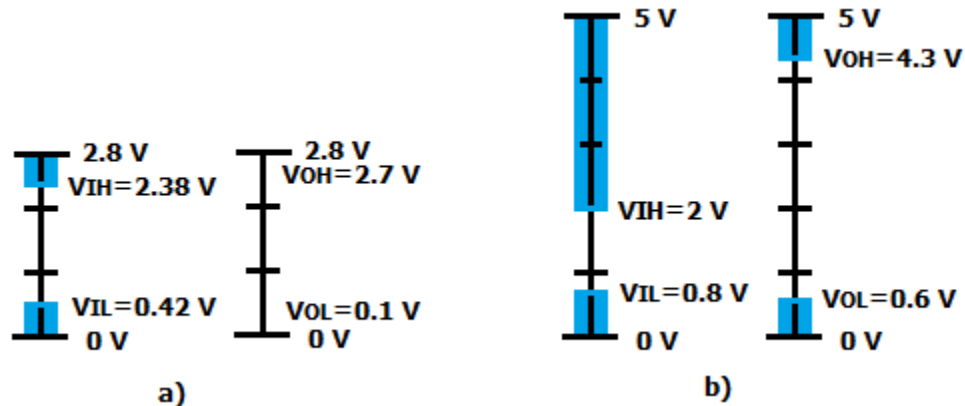


**Figura 2.11.** Conexión básica entre el PIC y el módulo SIM900.

Fuente: autor

A pesar de esta simple conexión, en la práctica este canal es muy sensible al ruido que puede provenir de la distribución de las pistas de la PCB o cuando en el microcontrolador se dejan pines de entrada no conectados o del propio módulo GSM/GPRS. Debido a esto puede ingresar ruido al sistema haciendo que aparezcan caracteres no deseados en el canal de comunicación o causando mal funcionamiento del PIC.

Por otro lado los niveles de voltaje lógicos entre módulos también puede ser un problema para establecer una adecuada comunicación.



**Figura 2.12.** Entradas y salidas lógicas. a) Módulo GSM/GPRS y b) PIC16F688

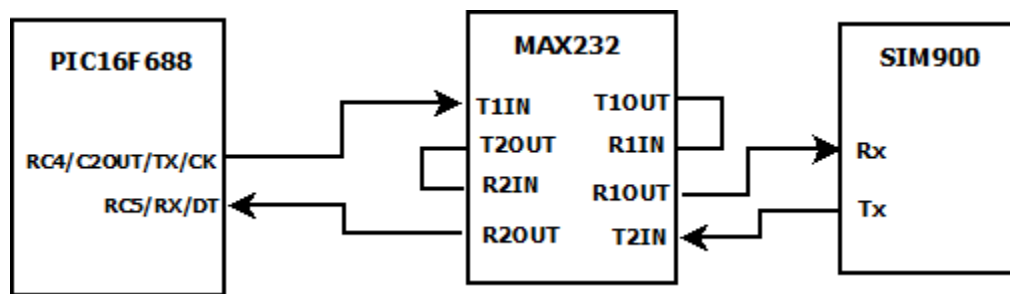
Fuente: autor

Como se observa en la figura 2.12 el rango de niveles lógicos es diferente entre ambas tecnologías. Como se puede notar las salidas del SIM900 son completamente compatibles con las entradas del PIC. Sin embargo las salidas del



PIC pueden generar incompatibilidad de niveles en algún momento, por ejemplo si la salida  $V_{OL}$  del PIC es 0.6 V generaría un estado inválido en el módulo SIM900 lo cual podría generar un estado inesperado durante la comunicación y eso se agravaría con un alto nivel de ruido en el sistema.

Es por esto que se conectará una interfaz o driver entre ambos módulos, el MAX232. Por lo general este módulo se usa como driver entre tecnologías TTL/COMS y RS232, sin embargo aquí se utilizará como un driver TTL/CMOS a TTL/CMOS con el objetivo ya planteado de mejorar la compatibilidad de niveles lógicos entre el PIC y el SIM900. De este modo la conexión entre PIC, MAX232 y SIM900 queda como se ve en la figura 2.13.



**Figura 2.13.** Interfaz entre PIC y SIM900.

Fuente: autor

La velocidad de transferencia de datos dependerá de la velocidad del módulo GSM/GPRS. Este módulo tiene soportados los siguientes baud rates:

1200, 2400, 4800, 9600, 19200, 38400, 57600 y 115200.

Con el oscilador interno del PIC configurado a 8 MHz se puede utilizar la máxima velocidad del SIM900 que es 115200 baudios. Sin embargo dado que el dato a transferir no es muy pesado se configurará la comunicación entre ellos a 9600 baudios con el PIC a 4 MHz.

Finalmente con el dato colocado en el módulo GSM/GPRS, este lo envía a la estación base más cercana de la red telefónica. El dato es enviado por GPRS

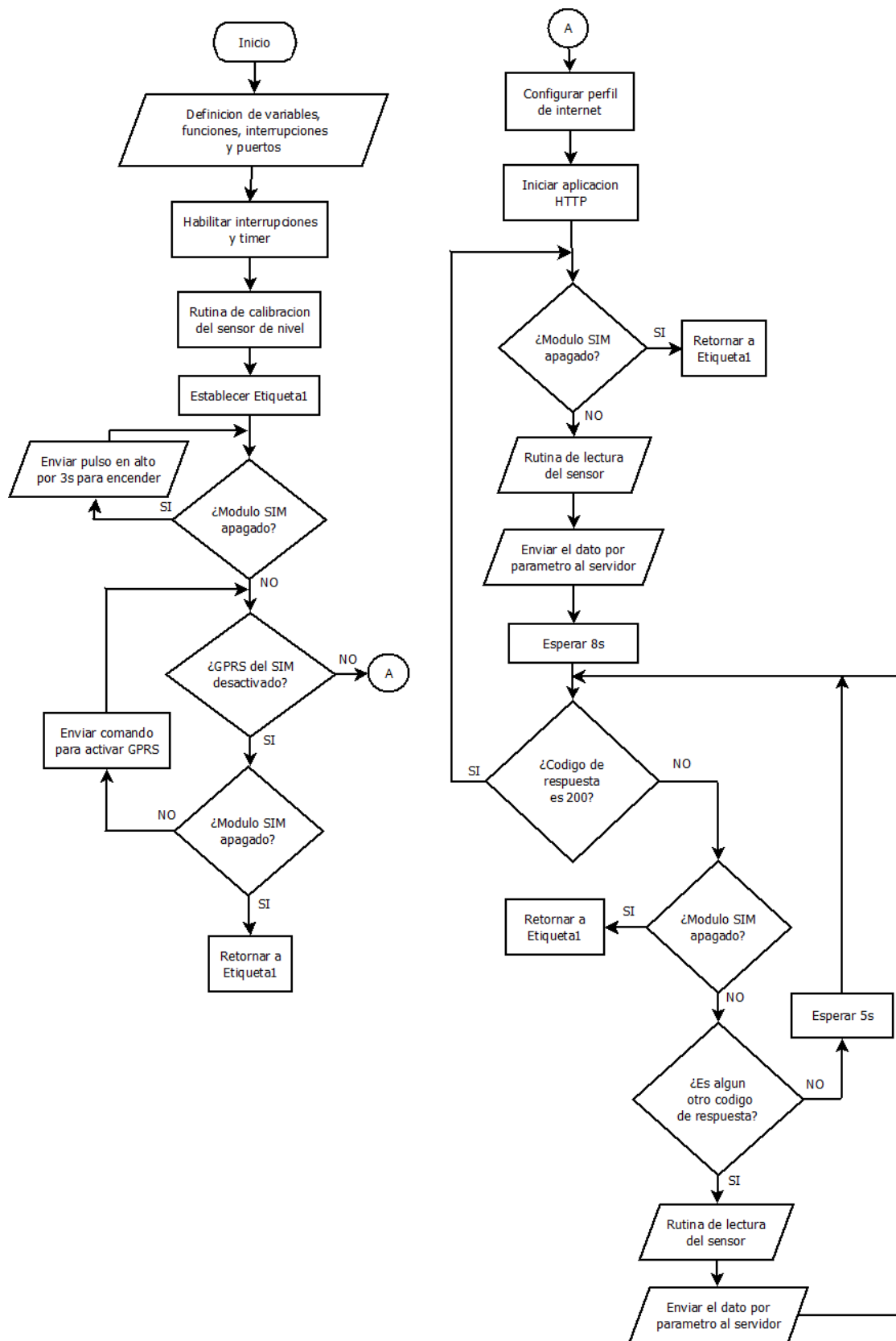
directamente al servidor web alojado en el web hosting gracias a la lista embebida TCP/IP que hace uso de la aplicación HTTP todo dentro del módulo SIM900.

Como se puede notar la comunicación y la transmisión entera está integrada en el módulo GSM/GPRS.

### **2.2.5 DISEÑO DEL SISTEMA DE CONTROL**

Teniendo el diseño del DAQ y el sistema de comunicación estamos en la posición de diseñar el sistema de control que comprende el proceso de lectura y la transferencia de datos hacia el módulo GSM/GPRS. No se puede decir, en la parte de control, sobre subir datos hacia el servidor porque esto lo hace en última instancia el módulo de transmisión.

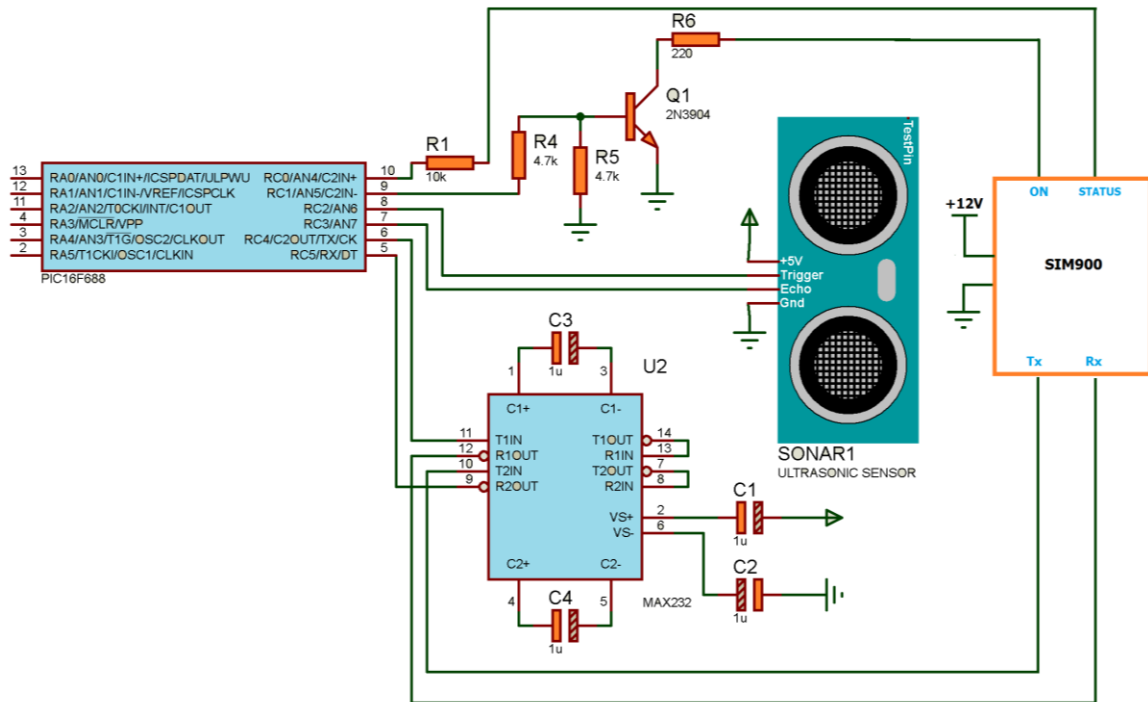
Para el diseño del sistema vamos a considerar el flujograma de la figura 2.14, el cual ha sido desarrollado para el uso del PIC16F688. En este no se entrará en detalles sobre cada una de las variables, directivas, rutinas o interrupciones que se van a utilizar sino más bien está referida a la funcionalidad de sistema.



**Figura 2.14.** Flujograma del sistema. Fuente: autor.

### 2.2.5.1 SIMULACION DEL SISTEMA DE COMUNICACION Y CONTROL

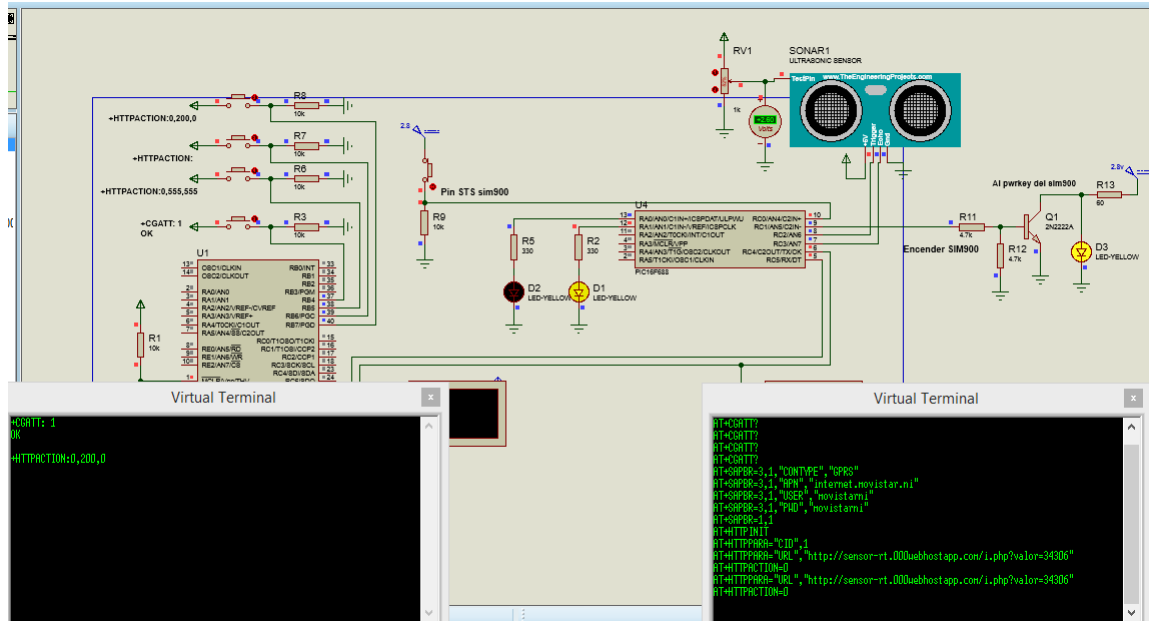
El diseño del sistema a nivel de circuito se muestra en la figura 2.15. En esta figura no se incluye el circuito correspondiente a la fuente de alimentación, el cual será tratado en apartados posteriores.



**Figura 2.15.** Diseño de circuito del sistema de control y comunicación..

Fuente: autor

En la figura 2.16 se observa la simulación corriendo en el programa Proteus 8.1. Como puede verse no se ha incluido el MAX232 porque a nivel de simulación las condiciones son ideales y no hay ruido o incompatibilidad de nivel de voltaje que afecte la comunicación y además que si se utilizan sus dos canales, en simulación, no funciona la comunicación.



**Figura 2.16.** Simulación del sistema en Proteus.

Fuente: autor

Al lado izquierdo se simula el módulo GSM/GPRS utilizando un PIC16F877A<sup>12</sup>, esto es porque no existe un modelo oficial para el SIM900 que pueda simularse en proteus.

## 2.2.5.2 COMANDOS PARA LA CONFIGURACIÓN DE INTERNET

Antes que el microcontrolador envíe una serie de comandos de configuración de internet hacia el módulo GSM/GPRS se debe garantizar que este tenga colocada una tarjeta sim en el módulo SIM900. Este paso previo incluye la compra del saldo y la activación del paquete de internet para la red telefónica que se esté usando.

En la tabla 2.11 se muestran todos los comandos que el PIC debe enviar al módulo GSM/GPRS para configurar éste para su comunicación con el servidor Web.

**Tabla 2.11**

*Comandos AT y sus respuestas para establecer conexión con el servidor*

<sup>12</sup> El código de simulación para este PIC se encuentra en Anexo D.2.

Comando AT ( $\mu$ Controlador)	Descripción	Respuesta (SIM900)
AT+CGATT? (o bien AT+CGATT=1)	Solicitud de activación del GPRS del SIM900.	+CGATT: 1 OK  +CGATT: 0 OK
AT+SAPBR=3,1,"CONTYPE", "GPRS"	Define el tipo de conexión a GPRS.	OK
AT+SAPBR=3,1,"APN", "internet.movistar.ni"	Define el punto de acceso a la red. En este caso para movistar Nicaragua.	OK
AT+SAPBR=3,1,"USER", "movistarni"	Define el usuario para acceso a la red (Movistar Nicaragua).	OK
AT+SAPBR=3,1,"PWD", "movistarni"	Define la contraseña para acceso a la red (Movistar Nicaragua).	OK
AT+SAPBR=1,1	Realiza la conexión a la red de datos.	OK
AT+HTTPIINIT	Habitación del modo HTTP	OK
AT+HTTTPARA="CID",1	Identificación del perfil del portador HTTP	OK
AT+HTTTPARA="URL", "http://direccion_del_servidor?parametro=valor"	Definir la URL a la dirección del servidor web.	OK
AT+HTTPACTION=0	Definir el método Action por el cual se realiza la petición al web server. En este caso, el método GET.	+HTTPACTION:0:xxx:yyy (xxx: código de respuesta del servidor. yyy: tamaño del dato)

**Nota.** Fuente: adaptado del data sheet del SIM900

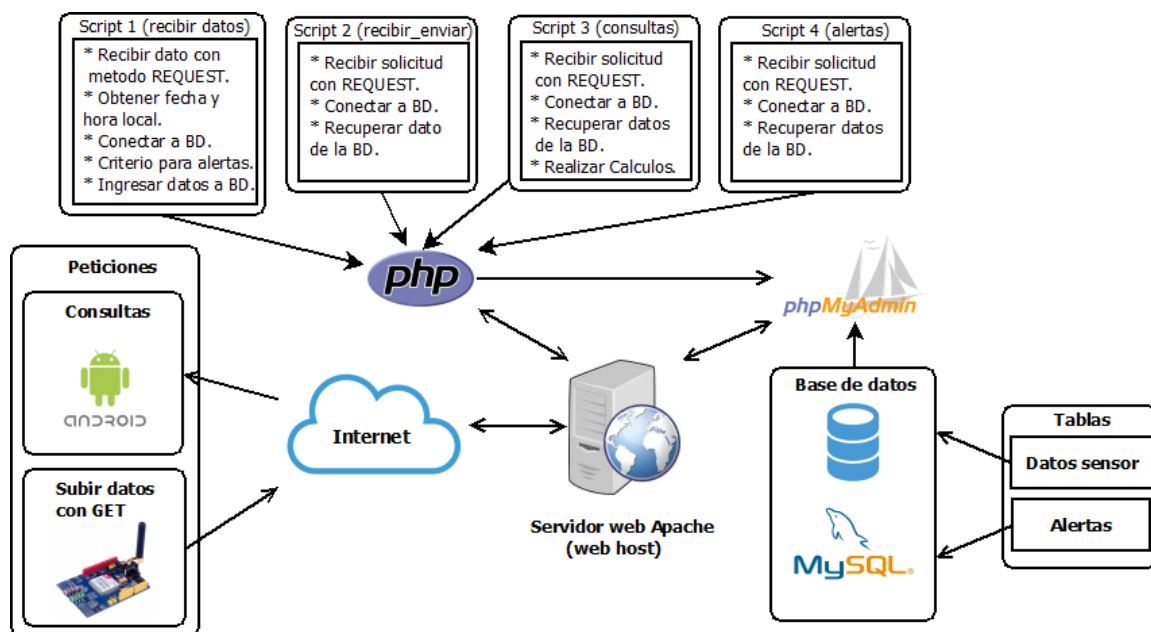
Cuando el módulo GSM/GPRS responde a una petición HTTPACTION, lo hace con un código<sup>13</sup> de respuesta HTTP. Si el dato se ha subido con éxito entonces el servidor web debe responder con el código 200.

## 2.2.6 DISEÑO DEL SERVIDOR WEB

El servidor web<sup>14</sup> se encontrará alojado en un web host el cual recibirá peticiones desde el módulo GSM/GPRS como también desde la aplicación Android. El sistema se muestra a través de los módulos de la figura 2.17.

<sup>13</sup> Ver en tabla B.4 en Anexos todos los posibles códigos de respuesta soportadas por el SIM900

<sup>14</sup> La creación del servidor en línea y su configuración se puede ver en Anexo F.1.



**Figura 2.17.** Diagrama de diseño del servidor web.

Fuente: autor

En diagrama de diseño se compone de varios bloques ya sea de hardware o software que componen el servidor web. La dirección de las flechas determina el flujo de los datos. El flujo puede ser unidireccional o bidireccional. Las peticiones al servidor o consultas a la base de datos no definen la dirección de las flechas.

### 2.2.6.1 PROCESO DE SUBIDA DE DATOS AL SERVIDOR

Si tomamos como punto de partida el módulo SIM900, el dato de nivel de agua inicialmente se envía desde este módulo hasta la nube y luego al servidor de nuestro web host. Una vez ahí se ejecuta el script (recibir datos) en PHP correspondiente para la recepción de este dato. Este script, toma la hora local actual de Nicaragua junto con el dato recibido y los almacena en la base de datos MySQL a través del gestor de bases de datos phpMyAdmin. La base de datos cuenta con dos tablas, una de datos y otra de alertas. Los datos de fecha y de nivel de agua siempre se guardan en la tabla de datos mientras que en la tabla alertas se guardan los datos de nivel que correspondan a niveles de agua muy elevados de acuerdo a un criterio evaluado en el script (recibir datos).

#### **2.2.6.2 PROCESO DE RECUPERACIÓN DE DATOS**

La recuperación de los registros de nivel y fecha/hora previamente guardados se realiza, en este caso, desde un dispositivo Android con acceso a internet que tenga instalada nuestra aplicación Android.

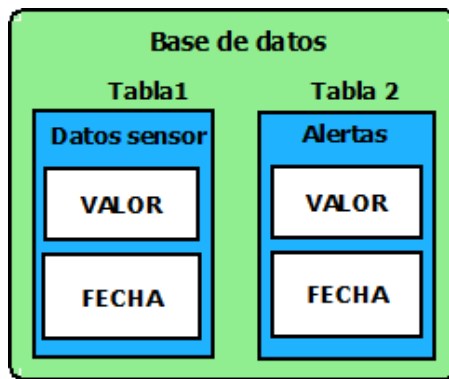
La función principal de la aplicación Android es realizar consultas, a un intervalo de tiempo, a la base de datos remota de modo que se tenga un flujo de datos. De esta forma se logra un monitoreo en tiempo real del nivel de agua a través de la aplicación Android.

Tomando como punto de partida la aplicación Android, esta realiza una consulta del dato de nivel de agua enviando una solicitud al servidor web en la nube. Seguidamente se activa el script asociado a las peticiones. Este recupera a través de phpMyAdmin el último registro ingresado en la tabla Datos de la base datos MySQL. Una vez recuperado el registro, el script lo envía a través de la nube a su destino, la aplicación Android.

#### **2.2.7 DISEÑO DE LA BASE DE DATOS**

La base de datos requiere de dos tablas, una para guardar los datos de fecha/hora y el valor leído del sensor de nivel y otra tabla para llevar un registro de los datos que están en el rango de alarmas. El esquema básico de la base de datos se ve en la figura 2.18.





**Figura 2.18.** Diagrama de la base de datos

Fuente: Autor.

La BD tendrá dos tablas nombradas “Datos sensor” y “Alertas”. Cada tabla tendrá dos atributos VALOR y FECHA/HORA. El tipo de dato para el atributo VALOR será de tipo int (entero) y para el atributo FECHA/HORA de tipo DATE/TIME (fecha/hora). El tipo de dato DATE/TIME, en phpMyAdmin, permite manejar datos de fecha y hora simultáneamente, de manera que no es necesario crear dos atributos para definir la fecha y la hora por separado.

La figura 2.19 muestra las tablas en el modelo lógico de la BD propio de phpMyAdmin. Se observa que no tienen relaciones (líneas de conexión) entre tablas ya que utilizamos una base bastante simple que consta de dos tablas no relacionadas.



**Figura 2.19.** Diseño lógico de la base de datos

Fuente: Autor

### 2.2.8 DISEÑO DE LOS SCRIPTS DEL SERVIDOR

Los scripts (programas) necesarios para la comunicación con el servidor escritos en PHP serán cuatro. A continuación se explica las funciones de cada uno.

Script 1:

- Se encarga de obtener el dato de nivel proveniente del módulo de transmisión. Este dato se obtiene con el método GET.
- Obtener, de la red, la fecha/hora de la zona de Nicaragua.
- Establecer conexión con la base de datos.
- Realizar cálculos aritméticos y operaciones lógicas de decisión.
- Realizar consultas SQL para inserción de registros a la tabla "Datos sensor" de la BD.
- Generar otras salidas de respuesta.

Script 2:

- Recibir solicitudes con el método REQUEST.
- Establecer conexión con la BD.
- Recuperar último dato insertado de la tabla "Datos sensor" de la BD.
- Generar salidas de respuesta.

Script 3:

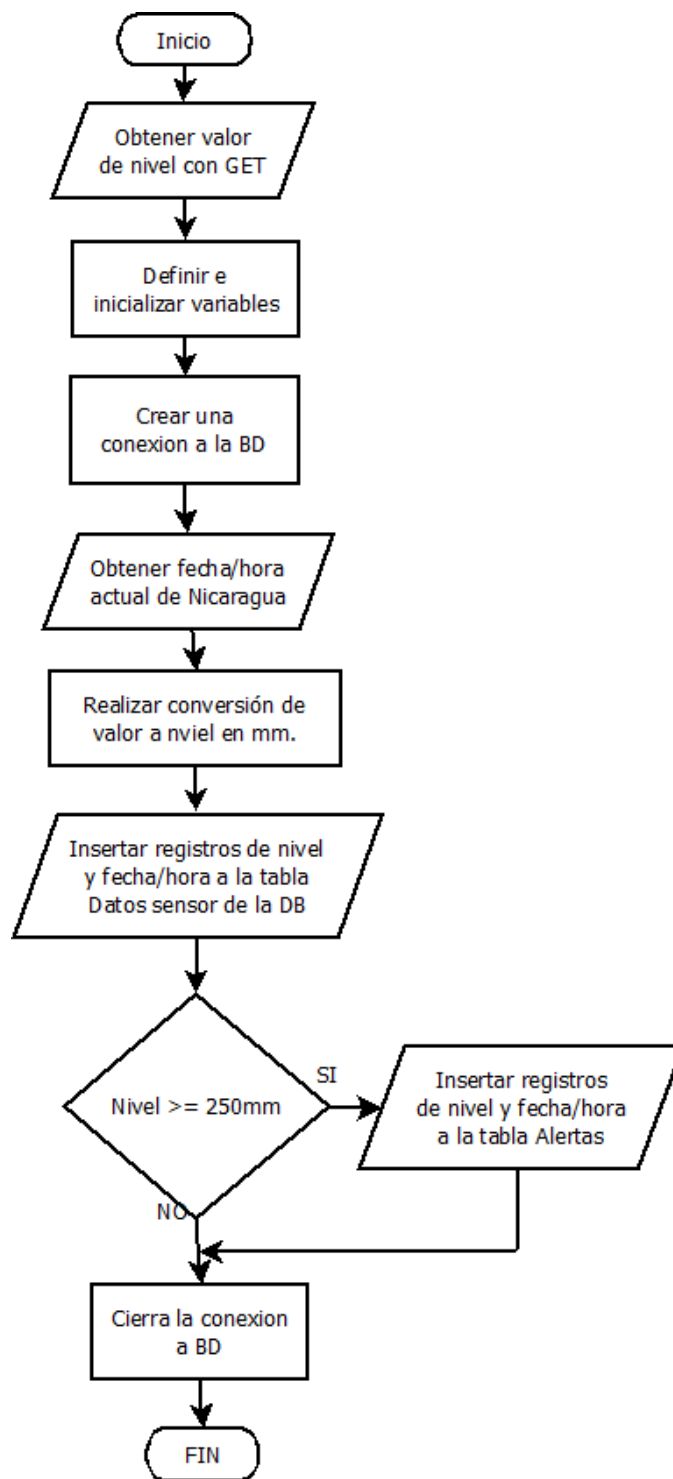
- Recibir solicitudes con el método REQUEST.
- Establecer conexión con la BD.
- Realizar operaciones lógicas de decisión.
- Realizar consultas SQL para recuperación de registros de la tabla "Datos sensor" de la BD.
- Generar otras salidas de respuesta.

Script 4:

- Recibir solicitudes con REQUEST.

- Establecer una conexión con la BD.
- Realizar consulta SQL para recuperar registros de la tabla "Alertas" de la BD.
- Generar otras salidas de respuesta.

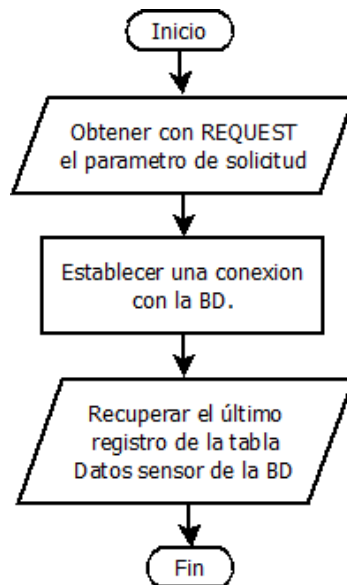
Los flujogramas para los scripts 1, 2, 3 y 4 se observan en las figuras 2.20, 2.21, 2.22 y 2.23 respectivamente.



**Figura 2.20.** Flujograma para el script 1.

Fuente: Autor.

El flujograma del script 1<sup>15</sup> se ejecutará solamente cuando el servidor recibe un dato enviado por parámetro desde cualquier conexión HTTP. En este caso la conexión la realiza el módulo GSM/GPRS usando HTTP1.1.



**Figura 2.21.** Flujograma para el script 2

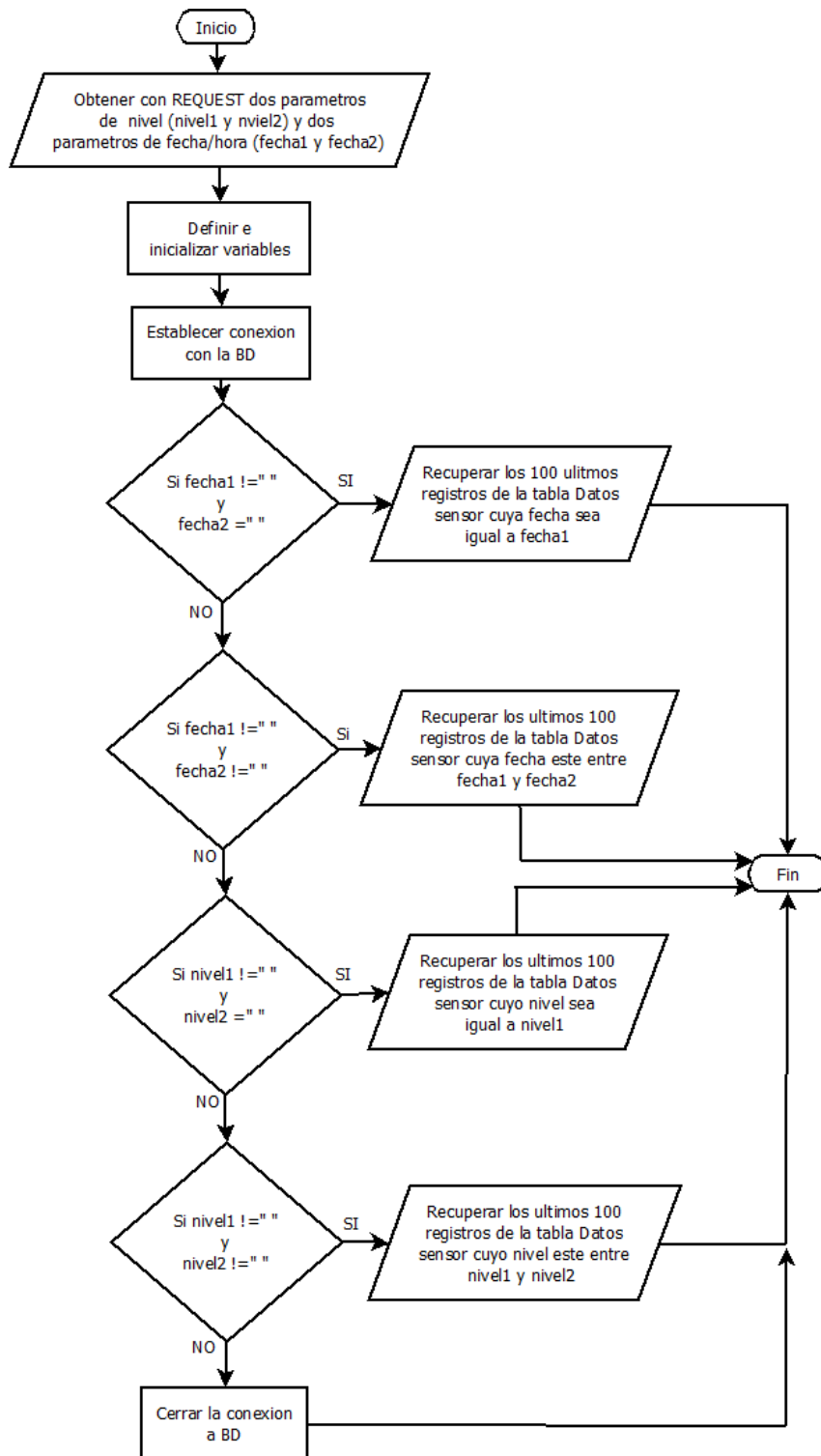
Fuente: Autor.

El script 2<sup>16</sup> se ejecuta solo cuando llega una solicitud al servidor dirigida a este script. El registro recuperado de nivel y fecha/hora es enviado a la App como respuesta a la solicitud realizada por la misma App.

---

<sup>15</sup> Ver código php en Anexo D.3

<sup>16</sup> Ver código php en Anexo D.4.



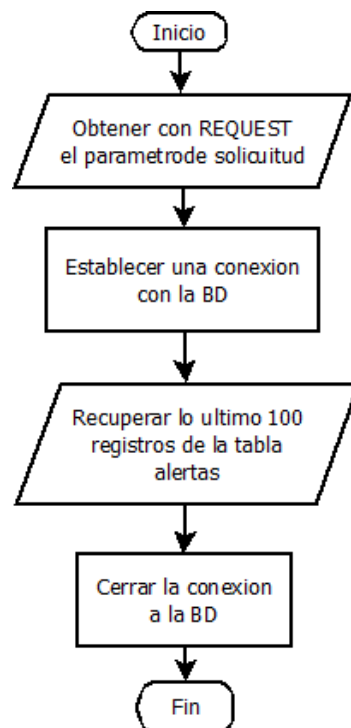
**Figura 2.22.** Flujograma para el script 3.

Fuente: Autor

Igual que para el script anterior, el script 3<sup>17</sup> se ejecutara solo cuando reciba cuatro datos enviados por parámetro al mismo tiempo. Este script está orientado a responder consultas del usuario. Este script recibe los datos de consulta realizados en la App y recupera los registros de la base de datos de acuerdo a cuatro posibles casos:

- Cuando fecha1 tiene un valor y fecha2 está vacío.
- Cuando fecha1 y fecha2 tienen valores.
- Cuando nivel1 tiene un valor y nivel2 está vacío.
- Cuando nivel1 y nivel2 tienen valores.

No se consideran los casos en los que las fechas estén vacías o los niveles estén vacíos ya que no se podrían hacer consultas SQL con datos vacíos. Además la App no debe permitir enviar las dos fechas vacías o los dos niveles vacíos.



**Figura 2.23.** Flujograma para el script 4.

Fuente: Autor.

---

<sup>17</sup> Ver código php en Anexo D.5.

El script 4<sup>18</sup> es más sencillo que los anteriores ya que con cada consulta de alertas realizadas por el usuario desde la App, este recupera los últimos 100 registros de alertas de la tabla Alertas.

## **2.2.9 DISEÑO DE LA APLICACIÓN ANDROID**

La aplicación Android tendrá varios procesos integrados en cuatro screens. Para entender mejor el funcionamiento de cada screen esta sección se subdividirá en otras secciones en la que se explica cada uno desde su flujograma. Si el flujograma es demasiado extenso como para alcanzar en una sola página cada flujograma se subdividirá a la vez en pequeños procesos o rutinas que serán mostrados como bloques expandidos del flujograma principal.

### **2.2.9.1 SCREEN PRINCIPAL**

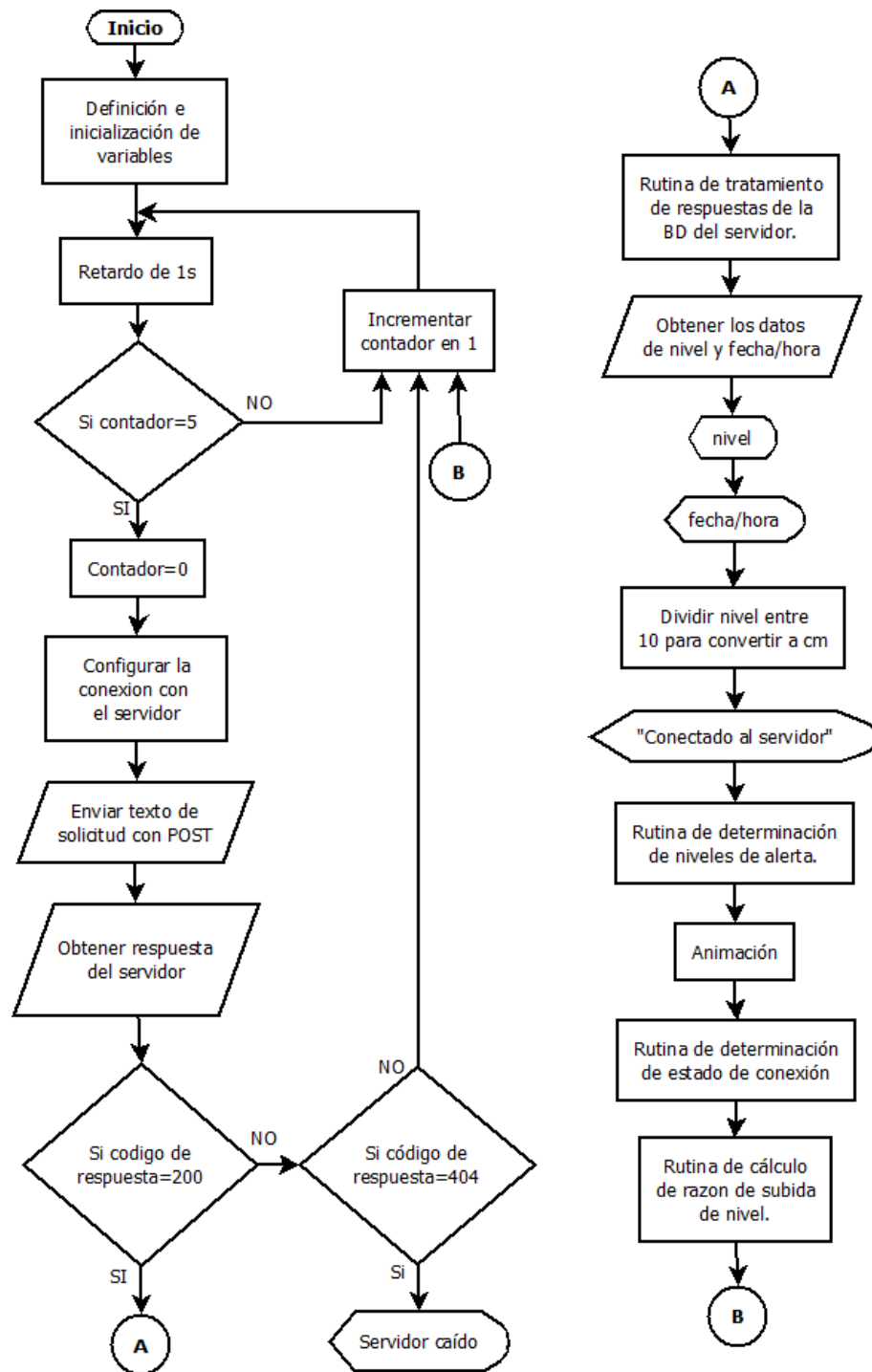
En general la screen principal<sup>19</sup> de la App, tiene la lógica del flujograma de la figura 2.24. En este caso el programa es algo extenso ya que cuenta con muchos procesos de configuración, decisión y aritméticos. Algunos bloques serán expandidos en una figura diferente para una mayor claridad.

---

<sup>18</sup> Ver código php en Anexo D.6.

<sup>19</sup> Ver código de bloque en Anexo E.1.





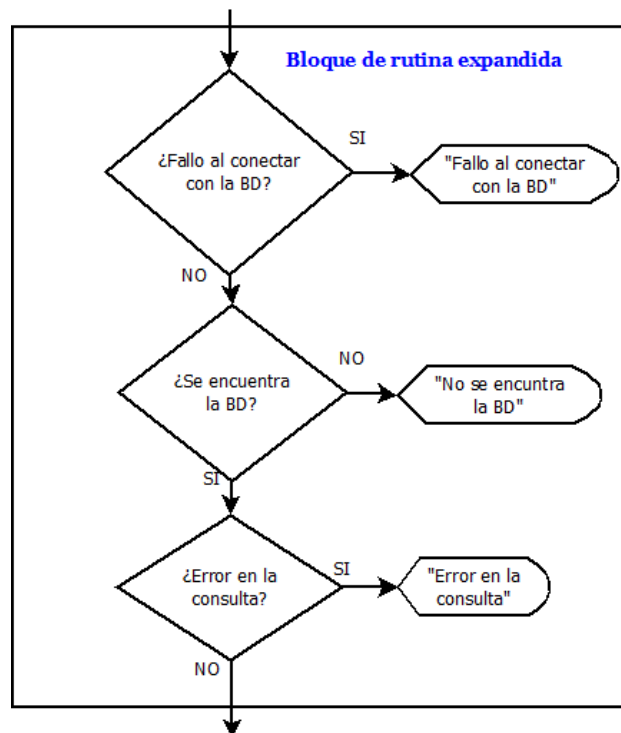
**Figura 2.24.** Flujograma de la screen principal de la App

Fuente: Autor

Como se observa, la App realiza una lectura cada 5 segundos. Aunque se podría poner a menos tiempo, con este intervalo se evita realizar demasiadas peticiones

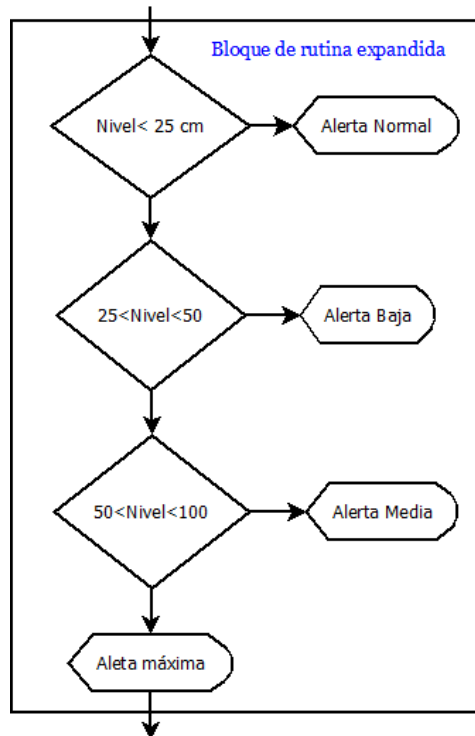
al servidor en corto tiempo, previniendo entre otras cosas el aumento del tráfico hacia y desde el servidor. Además este intervalo de lectura es apropiado para una correcta lectura en tiempo real, no es muy largo o muy corto, además prevenimos la lectura de un mismo dato dos o más veces.

Debido a que el flujograma es un poco extendido las rutinas empaquetadas en los bloques se expandirán para explicarse en detalle en las subsiguientes figuras.



**Figura 2.25.** Rutina de tratamiento de respuestas de la BD del servidor.

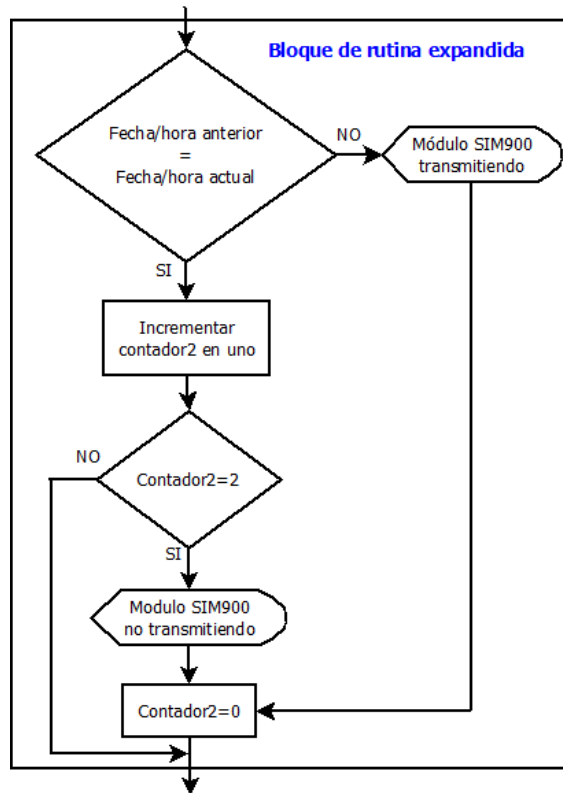
**Nota.** Esta rutina corresponde al bloque de rutina de tratamiento de respuestas de la BD del servidor de la screen principal. Fuente: Autor



**Figura 2.26.** Rutina de determinación de niveles de alerta.

**Nota.** Esta rutina corresponde al bloque de rutina de determinación de niveles de alerta de la screen principal. Fuente: Autor

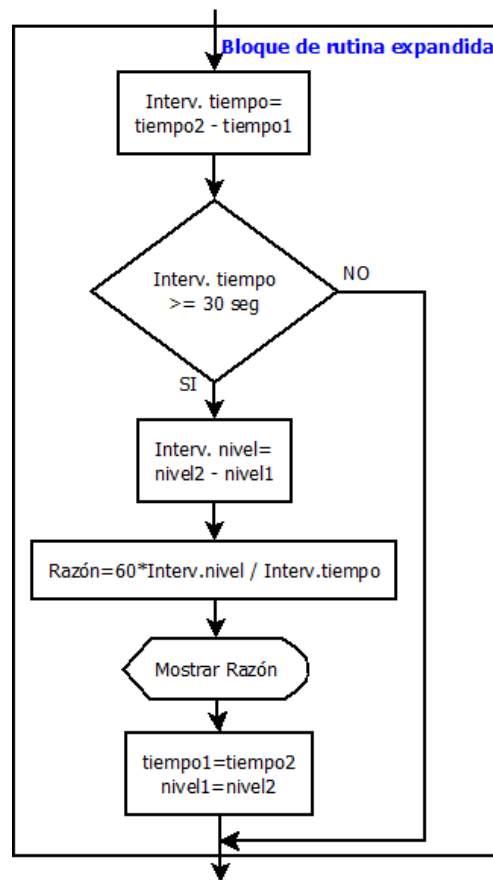
Para determinar si el módulo GSM/GPRS (SIM900) se encuentra transmitiendo es necesario saber si éste ha subido datos al servidor. Si la fecha/hora de lectura del nivel de agua es el mismo que la lectura anterior tres veces seguidas (15 segundos), eso significa que el modulo no se encuentra transmitiendo. En general la App por medio de la screen principal realiza una lectura cada 5 segundos por lo tanto el contador a utilizar debe llegar hasta 2, iniciando de 0, para contar esos 15 segundos. La determinación del estado de conexión de SIM900 se muestra en la figura 2.27.



**Figura 2.27.** Rutina de determinación de estado de conexión

**Nota.** Esta rutina corresponde al bloque de rutina de determinación de estado de conexión de la screen principal. Fuente: Autor.

El cálculo de la razón de subida del nivel de agua se hará con un intervalo de tiempo de 30 segundos. El resultado estará dado en mm/s, así que será multiplicado por 60 para convertirlo a mm/min para tener una medida más sensible. La razón será igual a dividir una variación de nivel entre una variación de tiempo para ese cambio de nivel. De este modo se deben tener cuatro variables, dos variables de nivel para calcular la diferencia de nivel y dos variables temporales para la diferencia de tiempo. Para restar las variables temporales se debe usar el formato de hora UNIX, con el propósito de poder operar con variables enteras en la App. La figura 2.28 muestra la lógica de este bloque.



**Figura 2.28.** Rutina de cálculo de razón de subida de nivel.

**Nota.** Esta rutina corresponde al bloque de rutina de cálculo de razón de subida de nivel de la screen principal. Fuente: Autor.

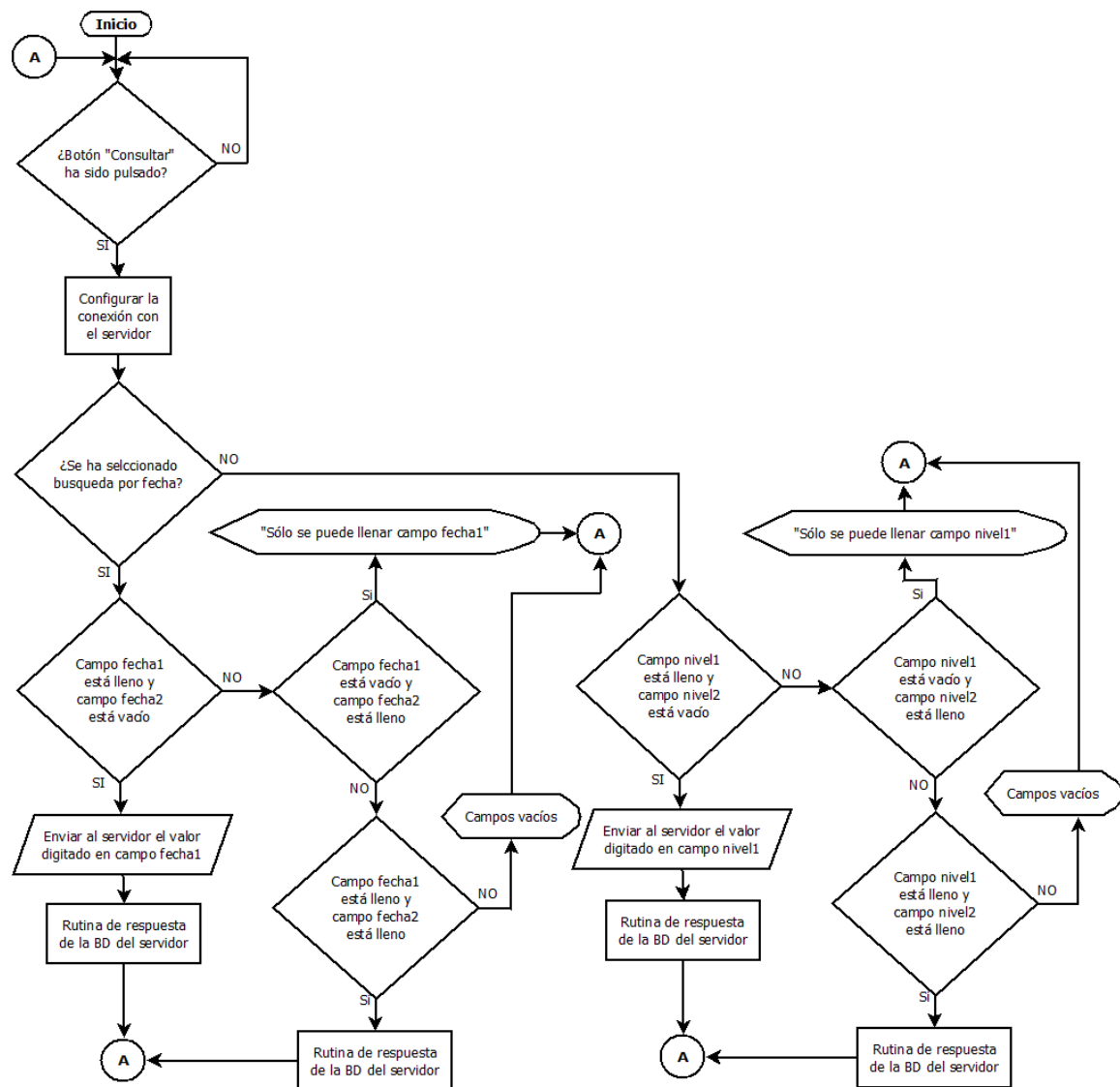
La App también debe contar con botones que permitan llevar a las demás pantallas o bien para salir de la misma. El código de bloque en MIT App Inventor se encuentra en anexo E.

### 2.2.9.2 SCREEN DE CONSULTAS

Las consultas se realizan por fecha/hora o por nivel. El formato fecha/hora ingresado en la screen consultas<sup>20</sup> debe ser año-mes-día hora:minuto:segundo. Ejemplo: 2019-01-01 23:59:00. La hora está en formato 24 horas dado que es el formato generado por PHP cuando obtiene la fecha/hora de la zona de Nicaragua.

<sup>20</sup> Ver código de bloque en Anexo E.2.

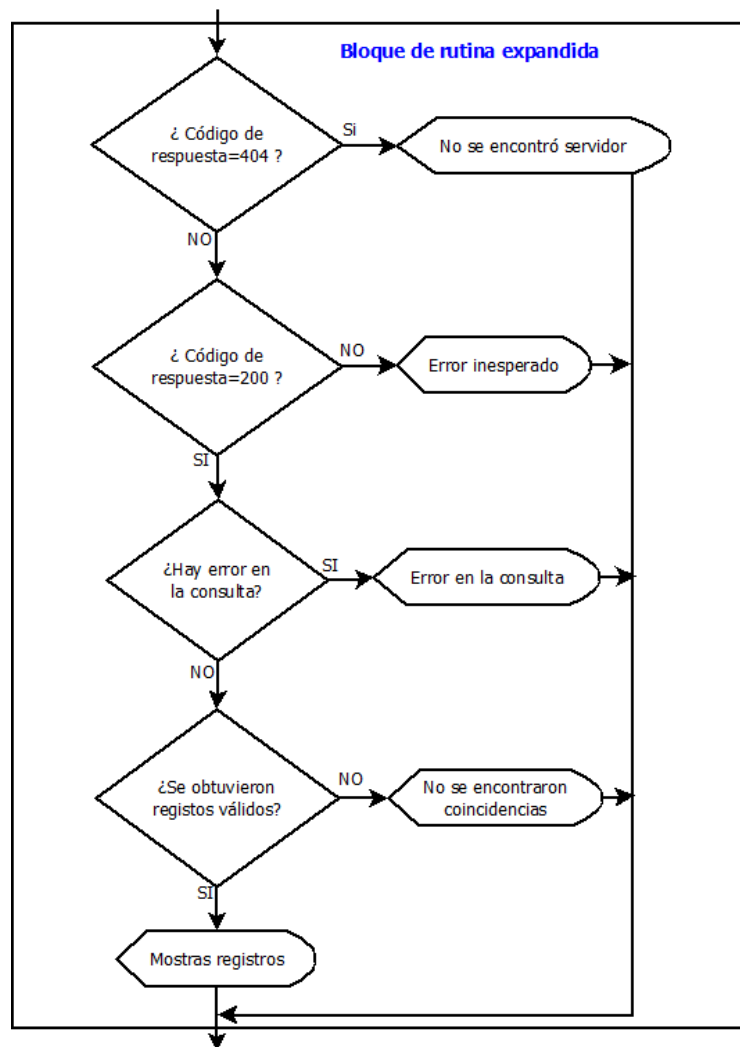
Cuando la consulta es por fecha/hora se puede hacer una búsqueda individual o entre fechas específicas incluyendo los segundos. En el caso del nivel es lo mismo y la respuesta del servidor estará dado en milímetros dado que así fue almacenado en la BD del servidor. Luego en la App se puede hacer una conversión a otra medida. El flujograma correspondiente se ve en la figura 2.29.



**Figura 2.29.** Flujograma para la screen Consultas.

Fuente: Autor

El servidor debe responder de acuerdo al criterio de consulta solicitado ya sea de fecha/hora o de nivel. El script en el servidor al cual deben ir dirigidas éstas solicitudes es “consulta.php” para acceder a los registros de la BD. El bloque expandido de la rutina de respuesta de la BD del servidor se ve en la figura 2.30.



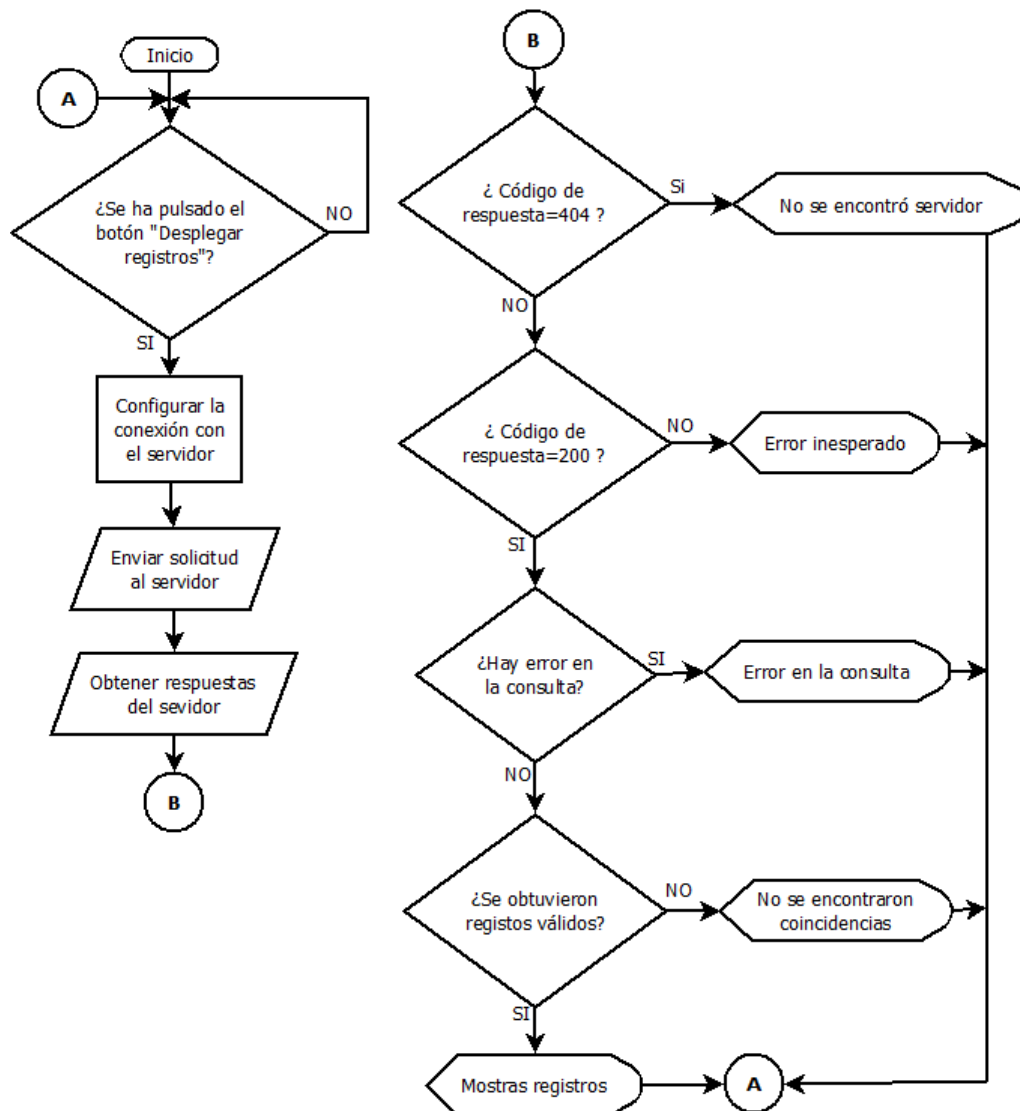
**Figura 2.30.** Rutina de respuesta de la BD del servidor.

Fuente: Autor

Los datos recuperados a mostrarse serán los últimos 100 registros de la BD en el servidor. Además se deben codificar en bloque los botones de ir atrás y otras procedimientos, Estas funciones adicionales de pueden ver en código de bloque en Anexos E.2.

### 2.2.9.3 SCREEN DE ALERTAS

La lógica para el flujograma de esta screen<sup>21</sup> es similar a la rutina de respuesta de la BD del servidor en la screen Consultas, salvo que aquí no se definen criterios de búsqueda, solamente se recuperan los 100 últimos registros de alarma de la BD por medio del script “alertas.php” alojado en el servidor. La lógica para este flujograma se observa en la figura 2.31.



**Figura 2.31.** Flujograma para la screen alertas.

Fuente: Autor

<sup>21</sup> Ver código de bloque en Anexo E.3.



## 2.2.10 DISEÑO DEL SISTEMA FOTOVOLTAICO

Para el diseño de este sistema primero debemos conocer el consumo teórico total que tendrá el sistema, que será igual a la suma de las cargas que aporten todos los elementos consumidores del sistema. En la tabla 2.12 se muestra el consumo de corriente de los componentes del sistema.

**Tabla 2.12**

*Cargas del sistema*

Componente	Consumo (mA)	Consumo (mW)
$\mu$ controlador PIC16F688	2	10
Shield GSM/GPRS. SIM900 (3.8V)+Regulador interno*	270	1026+2133
MAX232	8	40
Controlador de carga	10	120 (a 12 V)
Otros elementos	3	15
Total		3344

**Nota.** \* El regulador interno del shield GSM/GPRS consume  $270\text{mA} \cdot (12\text{V} - 4.1\text{V}) = 2133\text{ mW}$

Fuente: Autor

### 2.2.10.1 CALCULO DE LA CAPACIDAD DE LA BATERIA

Para el cálculo de la batería debemos tener en cuenta que ésta debe suministrar energía 24h/día, los días de autonomía y la profundidad de descarga, entonces los cálculos serían los siguientes:

$$\text{Energía/día} = 3.344\text{ W} \times 24\text{ horas/día} = 80.256\text{ Wh/día}$$

Dado que el voltaje de entrada es 12 V, la carga diaria es:

$$\text{Carga/día} = \frac{80.256\text{ Wh/día}}{12\text{ V}} = 6.688\text{ Ah/día}$$

Luego la capacidad de la batería está dada por:

$$C = \frac{C_{pre} * D_{aut}}{P_{des}} \quad (Ec. 2.4)$$

Donde,

$C$ : Capacidad de la batería

$C_{pre}$ : Consumo diario calculado.

$D_{aut}$ : Dias de autonomía.

$P_{des}$ : Profundidad de descarga.

$$C = \frac{6.688 \text{ Ah/día} * 3 \text{ días}}{0.8} = 25 \text{ Ah}$$

Dado el alto costo de las baterías de ácido selladas, sobre todo las de 25 Ah a 30 Ah, que oscilan entre \$ 80 y \$ 100, se decidió adquirir una batería más pequeña de 9 Ah a un costo de \$ 30. Este cambio hará que el sistema solo pueda lograr un día, una hora y 50 minutos de autonomía. Sin embargo aún en días nublados el módulo fotovoltaico puede generar voltaje para cargar la batería de forma que el tiempo de autonomía se podría estirar un poco más.

#### **2.2.10.2 CALCULO DE LA CAPACIDAD DEL MODULO FOTOVOLTAICO**

Para el cálculo del panel debemos encontrar el HSP (Horas solar pico). Para Nicaragua las HSP en promedio son 6, lo que significa 6000 Wh/día. Sin embargo este valor puede ser más exacto para un punto en particular en Managua si lo encontramos por medio de la página de Global Solar Atlas perteneciente a World Bank Group. El cálculo se hace para el punto crítico seleccionado en el análisis del sistema con coordenada específica 12°06'51.7"N 86°13'33.8"W. De esta forma generamos la salida de la figura 2.32.



**Figura 2.32.** Parámetros de radiación solar para un punto crítico.

Fuente: Reporte generado de <https://globalsolaratlas.info/>

De los parámetros seleccionaremos el GTI (Global Tilted Irradiation) o radiación global inclinada. Este parámetro mide la cantidad de energía en kWh/día para un módulo fotovoltaico inclinado. El OPTA (Angulo optimo) debe ser 15° con un azimut de 180°, es decir con el modulo inclinado 15° orientado al sur. Entonces el HSP se calcula como:

$$HSP = \frac{5.663 \text{ kWh}/(m^2 * \text{día})}{1 \text{ kW}/m^2} = 5.663 \text{ h/día}$$

El módulo fotovoltaico debe recibir entonces 5.66 h de sol pico diario. Luego los amperios-hora diarios producido por el módulo serian:

$$I_{GEN*h/día} = HSP * I_{MAX} = 5.663 * I_{MAX}$$

Como el consumo diario del sistema es 6.688 Ah/día, entonces la producción diaria del módulo debe ser mayor que ésta. Por lo tanto:

$$5.663 * I_{MAX} > 6.688$$

$$I_{MAX} > 1.18 A$$

Esto significa que la potencia máxima de operación de un solo módulo o la aportación varios módulos debe ser:

$$V_{mp} * I_{MAX} > 1.18A * V_{nom} \quad \text{pero } V_{nom} = 12 V$$

$$P_{mp} > 14.16 W$$

La potencia del módulo debe ser mayor que 14.16 W. En el mercado local, los paneles más pequeños que han sido cotizados son de 5 W a \$ 9. Se requerirían como máximo 3 paneles de estos conectados en paralelo para suplir la demanda, pero esto hace que el costo de estos suba a \$ 27, Explorando el mercado extranjero se pueden hallar módulos fotovoltaicos más eficientes de 15 W pero a precios de hasta \$ 56 sin considerar los costos de envío a Nicaragua. Por lo tanto se ha optado por adquirir, en el mercado local, un módulo de 30 W a \$ 27. Este módulo ocupa mayor superficie que los demás, pero satisface la demanda de potencia a un precio razonable.

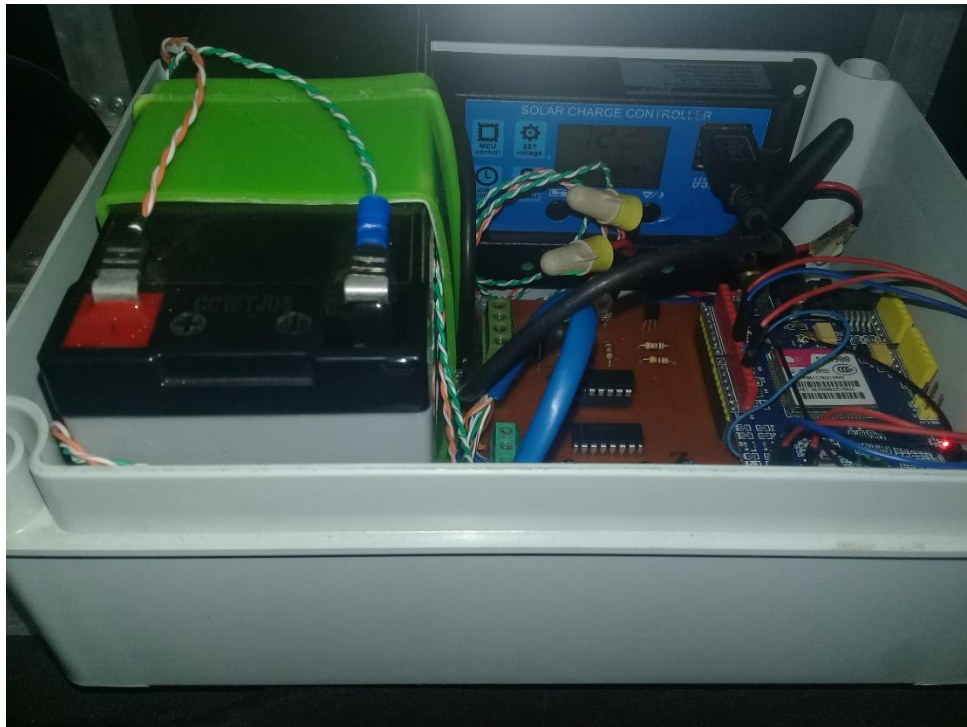
### **2.2.11 DISEÑO DE LAS PISTAS DE LA PCB**

El diseño de las pistas se hace de acuerdo al diagrama de la figura 2.33. Dado que algunos elementos, como los LEDS o el shield SIM900, no tienen un modelo para poder incluirlos en el diseño de las pistas, se han colocado terminales de bloque en su lugar. La figura 2.34 muestra el diseño de las pistas para el diagrama de la figura 2.33. En la pcb se ha integrado todo el sistema electrónico, el de comunicación y transmisión y el microcontrolador. El sensor no va montado en esta placa ya que este no debe ser de posición fija, sino que debe variar dependiendo de la instalación. Este sensor tiene su puerto de conexión por medio de cables a la pcb.



## 2.3. IMPLEMENTACION Y EVALUACIÓN

Una vez elaboradas las pistas de la pcb, montada la batería y el sensor y realizadas las interconexiones del sistema electrónico, la parte interna del hardware queda como se muestra en la figura 2.35. La figura 2.36 muestra la apariencia externa del hardware.



**Figura 2.35.** Sistema de monitoreo (hardware interno).

Fuente: autor.



**Figura 2.36.** Sistema de monitoreo (control y transmisión).

Fuente: autor

Dado que se ha propuesto ubicar el sistema en un punto geográfico específico del punto vulnerable, barrio anexo Jardines de Veracruz, no necesariamente implicaba que este se instalaría y pondría a funcionar en este sitio. Esto es porque se requiere un requisito legal, como un permiso de la alcaldía de Managua, además de los costos adicionales para la instalación y seguridad de la infraestructura misma que hizo superar nuestro presupuesto. De este modo la implementación del sistema se ha realizado en un sitio diferente, al menos en posición geográfica, sin condición de lluvia, con el módulo operando a unos 72 cm del suelo y con un recipiente con agua. Esto con el fin de realizar una prueba y evaluación del sistema.

### **2.3.1. PUESTA EN MARCHA DEL SISTEMA**

Lo primero que se hizo fue recargar la tarjeta sim del shield GSM/GPRS con un monto suficiente para activar el paquete de internet que ofrece la compañía Movistar. En este caso se usó un paquete de 600 Mb, que tiene un costo de C\$



50 válido por cuatro días. No es necesario retirar la tarjeta sim y colocarlo en un celular para activar el paquete, se puede hacer de forma remota usando la App “Mi Movistar” disponible en Google Play.

El sistema de control y transmisión no tiene botón de encendido puesto que debe transmitir 24 horas<sup>22</sup> diarias y no debe ser apagado. Sin embargo se puede apagar o encender desde el controlador de carga o desconectando la alimentación de 5V en el controlador solar. Una vez encendido el sistema, este inicia la rutina de lectura de nivel de referencia. Esto consiste en tomar 20 lecturas de distancia con el sensor y tomar la moda de estas lecturas. Hecho esto, se toma la primera lectura de nivel y se configura el sistema para establecer conexión con internet y subir este dato al servidor. En la figura 2.37 se observa el sistema en funcionamiento con agua en un recipiente y en la figura 2.38 se observan los registros de nivel en milímetros y la fecha/hora en la que fueron subidos.



**Figura 2.37.** Sistema operando en situación simulada

Fuente: autor

---

<sup>22</sup> 23 horas usando nuestro web hosting.



VALOR	FECHA
0	2019-04-10 11:45:28
0	2019-04-10 11:45:37
0	2019-04-10 11:45:47
0	2019-04-10 11:45:57
0	2019-04-10 11:46:06
0	2019-04-10 11:46:15
0	2019-04-10 11:46:25
0	2019-04-10 11:46:35
0	2019-04-10 11:46:44
0	2019-04-10 11:46:53
0	2019-04-10 11:47:03
0	2019-04-10 11:47:12
0	2019-04-10 11:47:22
0	2019-04-10 11:47:32
0	2019-04-10 11:47:42
391	2019-04-10 11:47:50
395	2019-04-10 11:47:59
200	2019-04-10 11:48:00

**Figura 2.38.** Panel de phpMyadmin.

Fuente: autor.

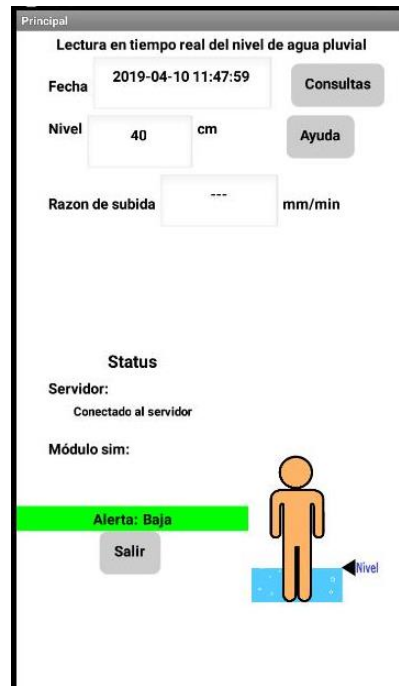
Como se observa en la figura 2.38 cada dato se subió aproximadamente a intervalos de 10 segundos. En la parte izquierda de la interfaz de phpMyAdmin se muestran las tablas nombradas como “alertas” y “datos\_sensor1” que contienen los registros nivel y fecha/hora. Los datos de nivel subidos fueron almacenados en la tabla “datos\_sensor1”. Si el dato de nivel subido supera los 250 mm éste se guarda en ambas tablas. En el servidor, los scripts que controlan éstas y otras acciones se muestran en la figura 2.39.

Nombre	Tamaño	Fecha	Permisos
consulta.php	1.7 kB	2018-11-27 18:19:00	-rw-r--r--
consulta2.php	2.7 kB	2018-11-27 23:17:00	-rw-r--r--
datos.csv	0.1 kB	2018-10-01 00:00:00	-rw-r--r--
datos_conexion.php	0.1 kB	2018-06-06 00:00:00	-rw-r--r--
Formulario_busqueda_proyecto.php	0.5 kB	2018-06-11 00:00:00	-rw-r--r--
i.php	1.4 kB	2019-04-03 17:18:00	-rw-r--r--
index2.php	1.0 kB	2018-05-18 00:00:00	-rw-r--r--
index3.php	2.9 kB	2018-06-06 00:00:00	-rw-r--r--
Insertar_Registro_fecha_hora.php	1.3 kB	2018-05-01 00:00:00	-rw-r--r--
pagina_busqueda_proyecto.php	1.5 kB	2018-06-12 00:00:00	-rw-r--r--
pagina_busqueda_proyecto1.php	1.6 kB	2018-06-14 00:00:00	-rw-r--r--
proyecto.php	1.0 kB	2018-08-20 00:00:00	-rw-r--r--

**Figura 2.39.** Archivos php en el servidor

Fuente: autor.

Desde la aplicación Android se pudo monitorear el nivel de agua, lo rápido que se incrementa su nivel, así como también hacer consultas sobre historial de registros. Para esto el dispositivo Android<sup>23</sup> tenía acceso a internet ya que la aplicación lo requiere para acceder al servidor en línea. La pantalla principal se observa en la figura 2.40.



**Figura 2.40.** App instalada en BlueStacks.

Fuente: autor.

### 2.3.2. EVALUACION DEL SISTEMA

Aunque el sistema no ha sido probado en condiciones de lluvia, la conexión con el servidor no se vio interrumpida, esto a menos que se diera un fallo de conexión para lo cual el microcontrolador está preparado reiniciando la conexión en ese caso.

La precisión del sensor, aunque depende exclusivamente del propio sensor, los redondeos y las conversiones de tipos de datos hicieron que el valor final

---

<sup>23</sup> Se usó el emulador de Android Bluestacks

mostrado en la aplicación Android tuviera un error no máximo de 1 cm, lo cual es aceptable.

Finalmente hay que hacer notar que el microcontrolador se vio afectado por el ruido generado por la propia circuitería sobre todo el proveniente de las pistas y del módulo GSM/GPRS. Así durante el proceso de la configuración inicial del módulo GSM/GPRS, el ruido se filtró al canal de comunicación entre éste módulo y el PIC, generando caracteres no deseados sobre el canal. Estos caracteres se agregaron a los comandos AT y a las respuestas del módulo GSM/GPRS. Esto en cierto momento generó que durante el inicio de la operación del microcontrolador, éste no configuró correctamente su módulo USART y el pin de recepción no se activó. Esto hizo que se tuviera que reiniciar manualmente el sistema más de una vez. Sin embargo cuando el microcontrolador se inició correctamente la conexión con el servidor se mantuvo durante todo el periodo de prueba, que fueron unas 2 horas.

## REFERENCIAS BIBLIOGRAFICAS

- Ali, M., & Causey, D. (2008). *PIC MICROCONTROLLER AND EMBEDDED SYSTEMS* (1ra ed.). New Jersey, United States: Pearson Prentice Hall.
- Amec Foster Wheeler. (2017). *PLAN MAESTRO DE ESCORRENTIA SUPERFICIAL (PMES) DEL MUNICIPIO DE MANAGUA*. Miami, Fl.
- Angulo, J., & Angulo, I. (2003). *Microcontroladores PIC* (3ra ed.). Madrid, España: McGraw-Hill.
- Christoffersen, J. (23 de febrero de 2017). *All about circuits*. Obtenido de <https://www.allaboutcircuits.com/projects/using-a-sim900a-to-send-sensor-data-to-a-website/>
- Colegio oficial de ingenieros de telecomunicación. (2002). *Energía Solar Fotovoltaica*. Madrid, España: Colegio Oficial de Ingenieros de Telecomunicación.
- Conde, C., & Saldaña, S. (2007). Cambio climático de América Latina y el caribe: Impactos, vulnerabilidad y adaptación. *Revista ambiente y desarrollo*, 23 (2).
- Díaz, J. (23 de marzo de 2015). Curso PHP MySQL. [Archivo de video]. Obtenido de [https://www.youtube.com/watch?v=I75CUdSJifw&list=PLU8oAlHdN5BkinrODGXToK9oPAlnJxmW\\_](https://www.youtube.com/watch?v=I75CUdSJifw&list=PLU8oAlHdN5BkinrODGXToK9oPAlnJxmW_)
- Domínguez, E., & Lozano, S. (2014). Estado del arte de los sistemas de alerta temprana en Colombia. *Revista de la Academia Colombiana de Ciencias Exactas, Físicas y Naturales*, 38 (148).
- Ente Vasco de la Energía (EVE). (2000). *La energía solar fotovoltaica en el País Vasco*. País Vasco: EVE.
- Gaddis, T., & Halsey, R. (2015). *Starting out with App Inventor for Android* (1st ed.). Halow, England: Pearson.
- García, E. (2008). *Compilador C CCS y simulador PROTEUS para Microcontroladores PIC* (1ra ed.). México D.F, México: Alfaomega.
- García, K. (7 de Abril de 2018). Managua tiene 59 puntos críticos ante desastres. *El Nuevo Diario*. Obtenido de <https://www.elnuevodiario.com.ni/nacionales/460404-managua-tiene-59-puntos-criticos-desastres/>
- Hostinger. (2019). *¿Qué es Apache? Descripción completa del servidor web Apache*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-apache/>
- INETER-COSUDE. (2005). *Inundaciones fluviales*. Managua: INETER-COSUDE.
- Kurose, J., & Ross, K. (2012). *Redes de computadoras: un enfoque descendente* (5ta ed.). Madrid, España: Pearson.

- Microchip. (2019). *PIC16F688 Data Sheet*. Obtenido de <http://ww1.microchip.com/downloads/en/DeviceDoc/41203E.pdf>
- Moreno, D., Quiñones, E., & Tovar, L. (2014). Los sistemas de alerta temprana, SAT, una herramienta para la prevención de desastres por inundación y efectos del cambio climático. *Revista Ciencias e Ingeniería al Día*, 9 (1).
- Morris, A. (2001). *Measurement and Instrumentation Principles* (3ra ed.). Oxford, Gran Bretaña: Butterworth-Heinemann.
- MySQL. (s.f.). *What's is MySQL?* Obtenido de <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- Pallás, R. (2003). *Sensores y acondicionadores de señal* (4ta ed.). Barcelona, España: Marcombo.
- Pavón, J. (2007). *Creación de un portal con PHP y MySQL* (3ra ed.). Madrid, España: RA-MA.
- PHP. (s.f.). *¿Qué es PHP?* Obtenido de <http://php.net/manual/es/intro-what-is.php>
- Sánchez, J. (2005). Análisis y estudio de redes GPRS (Trabajo de grado). Universidad Austral de Chile. Chile
- Tanenbaum, A., & Wetherall, D. (2012). *Redes de computadoras* (5ta ed.). Naucalpan de Juárez, México: Pearson.
- Villada, J. (2014). *Instalación y configuración del software de servidor Web*. Antequera, Málaga: IC Editorial.

## ANEXOS

### ANEXO A: ESPECIFICACIONES DE SENSORES, ICS Y MODULOS.

#### A.1. SENSOR ULTRASONICO HC-SR04

##### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time $\times$ velocity of sound (340M/S) / 2.

##### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

**Tabla A.1**

*Electric parameter*

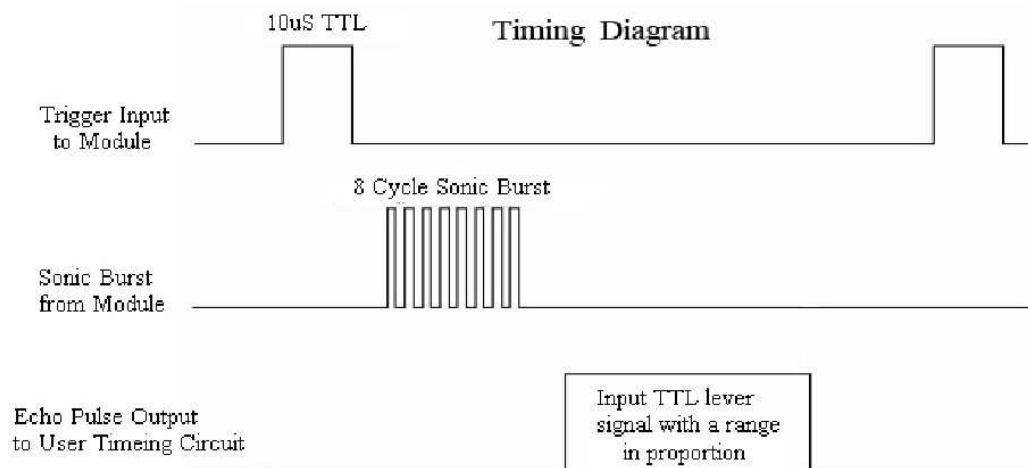
<b>Working Voltage</b>	<b>DC 5 V</b>
<b>Working Current</b>	<b>15mA</b>
<b>Working Frequency</b>	<b>40Hz</b>
<b>Max Range</b>	<b>4m</b>
<b>Min Range</b>	<b>2cm</b>
<b>MeasuringAngle</b>	<b>15 degree</b>
<b>Trigger Input Signal</b>	<b>10uS TTL pulse</b>
<b>Echo Output Signal</b>	<b>Input TTL lever signal and the range in proportion</b>
<b>Dimension</b>	<b>45*20*15mm</b>



**Figura A.1.** HC-SR04 pinout

## Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula:  $\mu\text{S} / 58 = \text{centimeters}$  or  $\mu\text{S} / 148 = \text{inch}$ ; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



**Figura A.2.** Timing diagram

## A.2. DRIVER/RECEPTOR MAX232

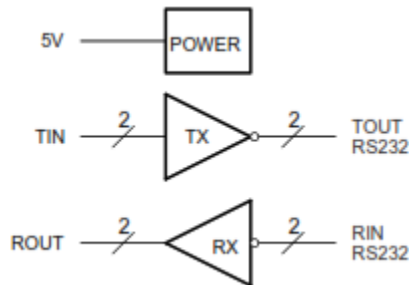
### Features

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0-μF Charge-Pump Capacitors

- Operates up to 120 kbit/s
- Two Drivers and Two Receivers
- $\pm 30$ -V Input Levels
- Low Supply Current: 8 mA Typical
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- $\mu$ F Charge-Pump Capacitors is Available With the MAX202 Device

## Applications

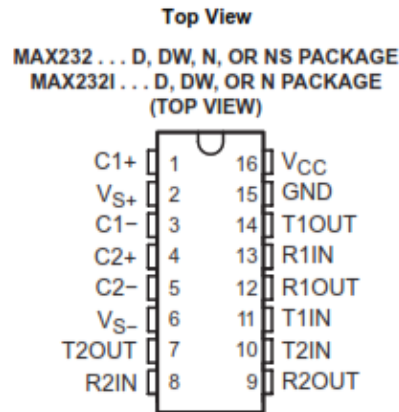
- TIA/EIA-232-F
- Battery-Powered Systems
- Terminals
- Modems
- Computers



**Figura A.3.** Simplified schematic

## Description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5V, and can accept  $\pm 30$ -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels.



**Figura A.4.** Pin configuration



**Tabla A.2***Pin functions*

PIN		TYPE	DESCRIPTION
NAME	NO.		
C1+	1	—	Positive lead of C1 capacitor
VS+	2	O	Positive charge pump output for storage capacitor only
C1-	3	—	Negative lead of C1 capacitor
C2+	4	—	Positive lead of C2 capacitor
C2-	5	—	Negative lead of C2 capacitor
VS-	6	O	Negative charge pump output for storage capacitor only
T2OUT, T1OUT	7, 14	O	RS232 line data output (to remote RS232 system)
R2IN, R1IN	8, 13	I	RS232 line data input (from remote RS232 system)
R2OUT, R1OUT	9, 12	O	Logic data output (to UART)
T2IN, T1IN	10, 11	I	Logic data input (from UART)
GND	15	—	Ground
V <sub>CC</sub>	16	—	Supply Voltage, Connect to external 5V power supply

**Tabla A.3***Recommended operating conditions*

		MIN	NOM	MAX	UNIT
V <sub>CC</sub>	Supply voltage	4.5	5	5.5	V
V <sub>IH</sub>	High-level input voltage (T1IN, T2IN)	2			V
V <sub>IL</sub>	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			±30	V
T <sub>A</sub>	Operating free-air temperature	MAX232		0	°C
		MAX232I		-40	

**Tabla A.4***Electrical characteristics- Driver*

PARAMETER		TEST CONDITIONS <sup>(1)</sup>		MIN	TYP <sup>(2)</sup>	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND	5	7		V
V <sub>OL</sub>	Low-level output voltage <sup>(3)</sup>	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND		-7	-5	V
r <sub>O</sub>	Output resistance	T1OUT, T2OUT	V <sub>S+</sub> = V <sub>S-</sub> = 0, V <sub>O</sub> = ±2 V	300			Ω
I <sub>OS</sub> <sup>(4)</sup>	Short-circuit output current	T1OUT, T2OUT	V <sub>CC</sub> = 5.5 V, V <sub>O</sub> = 0 V		±10		mA
I <sub>IS</sub>	Short-circuit input current	T1IN, T2IN	V <sub>I</sub> = 0			200	μA

(1) Test conditions are C1–C4 = 1 μF at VCC = 5 V ± 0.5 V

(2) All typical values are at VCC = 5 V, TA = 25°C.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

(4) Not more than one output should be shorted at a time.

**Tabla A.5***Electrical characteristics- Receiver*

PARAMETER		TEST CONDITIONS <sup>(1)</sup>	MIN	TYP <sup>(2)</sup>	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	R1OUT, R2OUT	I <sub>OH</sub> = -1 mA	3.5		V
V <sub>OL</sub>	Low-level output voltage <sup>(3)</sup>	R1OUT, R2OUT	I <sub>OL</sub> = 3.2 mA		0.4	V
V <sub>IT+</sub>	Receiver positive-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	1.7	2.4	V
V <sub>IT-</sub>	Receiver negative-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	0.8	1.2	V
V <sub>hys</sub>	Input hysteresis voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V	0.2	0.5	V
r <sub>i</sub>	Receiver input resistance	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	3	5	7 kΩ

(1) Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

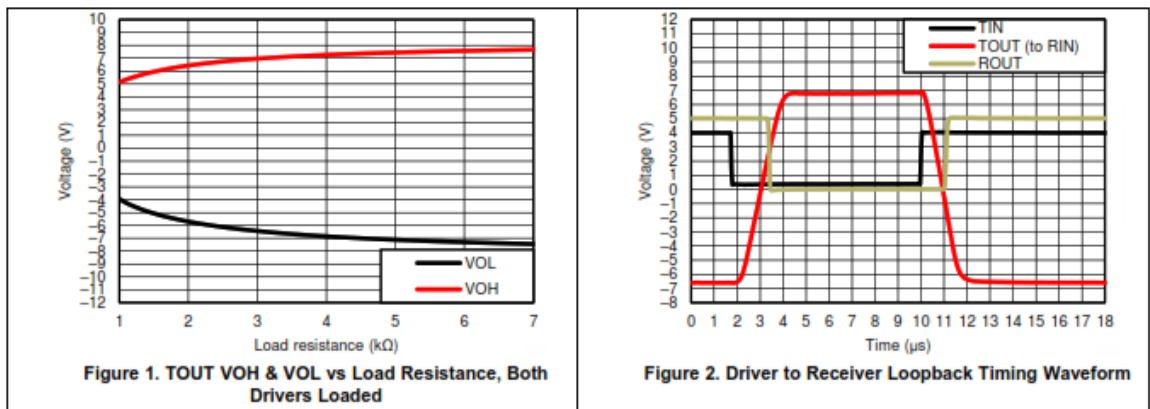
(2) All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

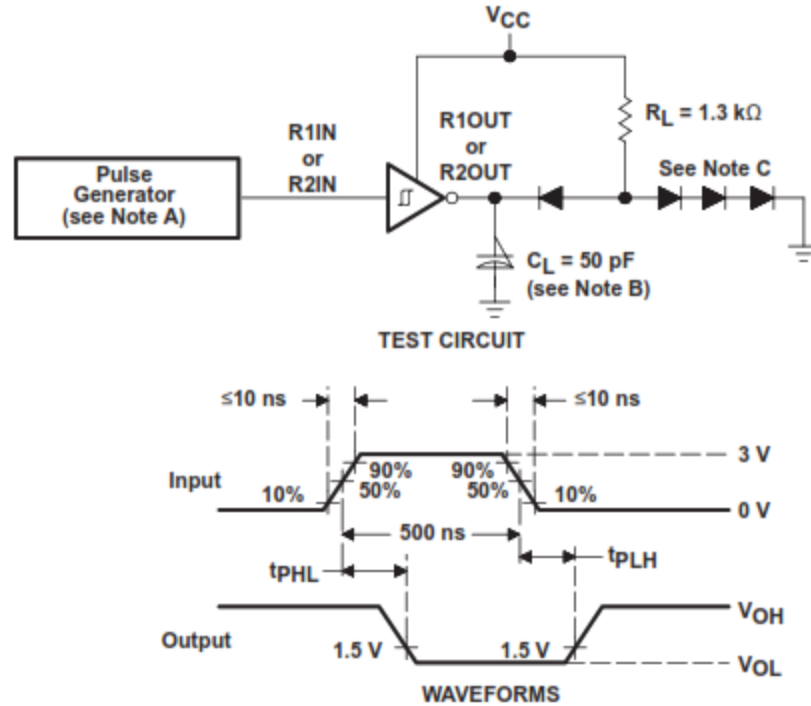
**Tabla A.6**

*Switching characteristics*

PARAMETER		TEST CONDITIONS <sup>(1)</sup>	MIN	TYP <sup>(1)</sup>	MAX	UNIT
SR	Driver slew rate	R <sub>L</sub> = 3 kΩ to 7 kΩ, see Figure 4			30	V/μs
SR(t)	Driver transition region slew rate	see Figure 5		3		V/μs
	Data rate	One TOUT switching		120		kbit/s
t <sub>PLH(8)</sub>	Receiver propagation delay time, low- to high-level output	TTL load, see Figure 3		500		ns
t <sub>PHL(8)</sub>	Receiver propagation delay time, high- to low-level output	TTL load, see Figure 3		500		ns

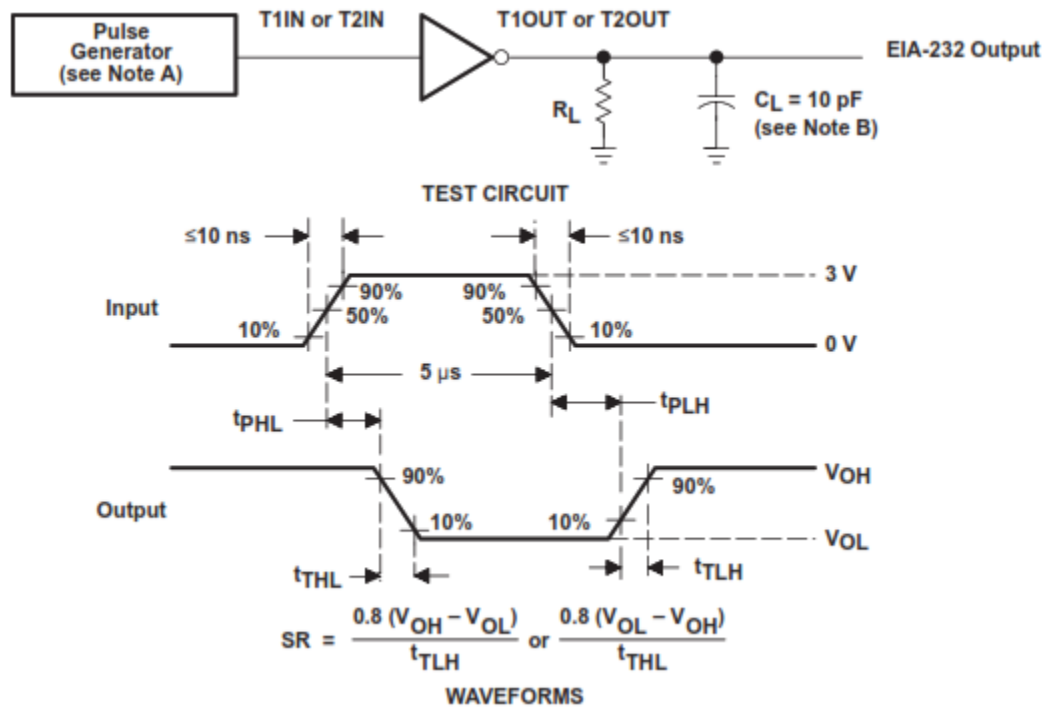


**Figura A.5.** *Typical characteristics*



**Figure A.6.** Receiver test circuit and waveform for  $t_{PHL}$  and  $t_{PLH}$  measurements

- A. The pulse generator has the following characteristics:  $Z_O = 50 \text{ }\Omega$ , duty cycle = 50%.
- B.  $C_L$  includes probe and jig capacitance.
- C. All diodes are 1N3064 or equivalent.



**Figure A.7.** Driver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements (5- $\mu\text{s}$  Input)

- A. The pulse generator has the following characteristics:  $Z_O = 50\ \Omega$ , duty cycle = 50%.  
 B. CL includes probe and jig capacitance.

**Table A.7**

*Function Table Each Driver<sup>(1)</sup>*

INPUT TIN	OUTPUT TOUT
L	H
H	L

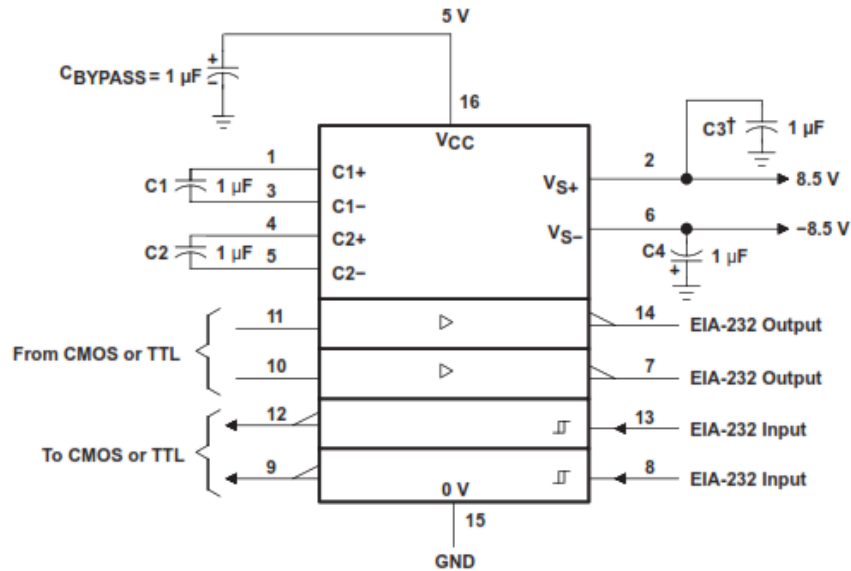
(1) H = high level, L = low level, X = irrelevant, Z = high impedance

**Table A.8**

*Function Table Each Receiver<sup>(1)</sup>*

INPUTS RIN	OUTPUT ROUT
L	H
H	L
Open	H

(1) H = high level, L = low level, X = irrelevant, Z = high impedance (off),  
 Open = disconnected input or connected driver off



**Figure A.8.** Typical operating circuit.

† C3 can be connected to  $V_{CC}$  or GND.

NOTES: A. Resistor values shown are nominal.

B. Nonpolarized ceramic capacitors are acceptable. If polarized tantalum or electrolytic capacitors are used, they should be connected as shown. In addition to the 1- $\mu$ F capacitors shown, the MAX202 can operate with 0.1- $\mu$ F capacitors.

## A.3. MICROCONTROLADOR PIC16F688

### A.3.1 DESCRIPCIÓN GENERAL DEL PIC16F688

#### High-Performance RISC CPU:

- Only 35 Instructions to Learn:
  - All single-cycle instructions except branches
- Operating Speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt Capability
- 8-level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes

#### Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to  $\pm 1\%$
  - Software selectable frequency range of 8 MHz to 125 kHz
  - Software tunable
  - Two-Speed Start-Up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide Operating Voltage Range (2.0V-5.5V)
- Industrial and Extended Temperature Range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with Software Control Option
- Enhanced Low-Current Watchdog Timer (WDT)
  - with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with Weak Pull-up or Input Only Pin
- Programmable Code Protection
- High-Endurance Flash/EEPROM Cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years

#### Low-Power Features:

- Standby Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11  $\mu$ A @ 32 kHz, 2.0V, typical
  - 220  $\mu$ A @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1  $\mu$ A @ 2.0V, typical

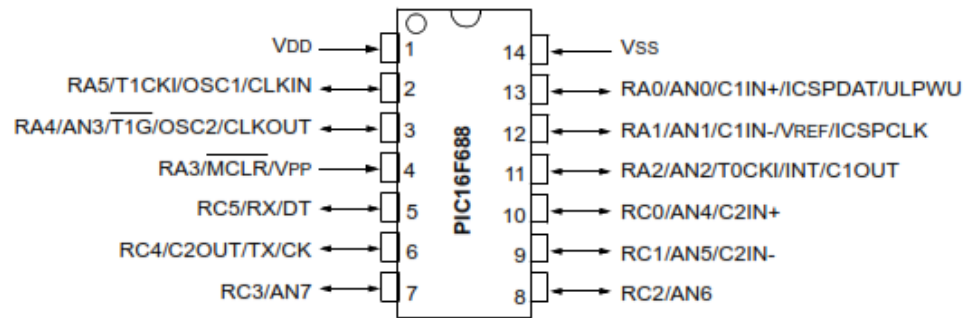
#### Peripheral Features:

- 12 I/O Pins with Individual Direction Control:
  - High-current source/sink for direct LED drive
  - Interrupt-on-change pin
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up
- Analog Comparator module with:
  - Two analog comparators
  - Programmable On-chip Voltage Reference (CVREF) module (% of VDD)
  - Comparator inputs and outputs externally accessible
- A/D Converter:
  - 10-bit resolution and 8 channels
- Timer0: 8-bit Timer/Counter with 8-bit Programmable Prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Timer1 Gate (count enable)
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator if INTOSC mode selected
- Enhanced USART Module:
  - Supports RS-485, RS-232, LIN 2.0/2.1 and J2602
  - Auto-Baud Detect
  - Auto-wake-up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via two pins

**Tabla A.9**

hardware summary

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)				
PIC16F688	4096	256	256	12	8	2	1/1



**Figura A.9.** Pin diagram (PDIP, SOIC, TSSOP)

**Tabla A.10****PIC16F688 PINOUT DESCRIPTION**

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN0	AN	—	A/D Channel 0 input
	C1IN+	AN	—	Comparator 1 input
	ICSPDAT	TTL	CMOS	Serial Programming Data I/O
	ULPWU	AN	—	Ultra Low-Power Wake-up input
RA1/AN1/C1IN-/VREF/ICSPCLK	RA1	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN1	AN	—	A/D Channel 1 input
	C1IN-	AN	—	Comparator 1 input
	VREF	AN	—	External Voltage Reference for A/D
	ICSPCLK	ST	—	Serial Programming Clock
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN2	AN	—	A/D Channel 2 input
	T0CKI	ST	—	Timer0 clock input
	INT	ST	—	External Interrupt
	C1OUT	—	CMOS	Comparator 1 output
RA3/MCLR/VPP	RA3	TTL	—	PORTA input with interrupt-on-change
	MCLR	ST	—	Master Clear w/internal pull-up
	VPP	HV	—	Programming voltage
RA4/AN3/T1G/OSC2/CLKOUT	RA4	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	AN3	AN	—	A/D Channel 3 input
	T1G	ST	—	Timer1 gate
	OSC2	—	XTAL	Crystal/Resonator
	CLKOUT	—	CMOS	Fosc/4 output
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	PORTA I/O w/prog pull-up and interrupt-on-change
	T1CKI	ST	—	Timer1 clock
	OSC1	XTAL	—	Crystal/Resonator
	CLKIN	ST	—	External clock input/RC oscillator connection
RC0/AN4/C2IN+	RC0	TTL	CMOS	PORTC I/O
	AN4	AN	—	A/D Channel 4 input
	C2IN+	AN	—	Comparator 2 input
RC1/AN5/C2IN-	RC1	TTL	CMOS	PORTC I/O
	AN5	AN	—	A/D Channel 5 input
	C2IN-	AN	—	Comparator 2 input
RC2/AN6	RC2	TTL	CMOS	PORTC I/O
	AN6	AN	—	A/D Channel 6 input
RC3/AN7	RC3	TTL	CMOS	PORTC I/O
	AN7	AN	—	A/D Channel 7 input
RC4/C2OUT/TX/CK	RC4	TTL	CMOS	PORTC I/O
	C2OUT	—	CMOS	Comparator 2 output
	TX	—	CMOS	USART asynchronous output
	CK	ST	CMOS	USART asynchronous clock
RC5/RX/DT	RC5	TTL	CMOS	Port C I/O
	RX	ST	CMOS	USART asynchronous input
	DT	ST	CMOS	USART asynchronous data
VSS	VSS	Power	—	Ground reference
VDD	VDD	Power	—	Positive supply

**Legend:** AN = Analog input or output    CMOS = CMOS compatible input or output    OC = Open collector output

TTL = TTL compatible input    ST = Schmitt Trigger input with CMOS levels

HV = High Voltage    XTAL = Crystal

### **A.3.2. MODULO EUSART EN EL PIC16F688**

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock polarity in synchronous modes

#### **A.3.2.1 MODO ASINCRONO EN EL EUSART**

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a V<sub>OH</sub> mark state which represents a '1' data bit, and a V<sub>OL</sub> space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is 8 bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

#### **A.3.2.2 TRANSMISIÓN ASÍNCRONA EN EL EUSART**

The EUSART transmitter block diagram is shown in Figure 10-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.



## Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state. Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

**Note 1:** When the SPEN bit is set, the RX/DT I/O pin is automatically configured as an input, regardless of the state of the corresponding TRIS bit and whether or not the EUSART receiver is enabled. The RX/DT pin data can be read via a normal PORT read but PORT latch data output is precluded.

**2:** The TXIF transmitter interrupt flag is set when the TXEN enable bit is set.

## Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one TCY immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

## Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

## TSR Status

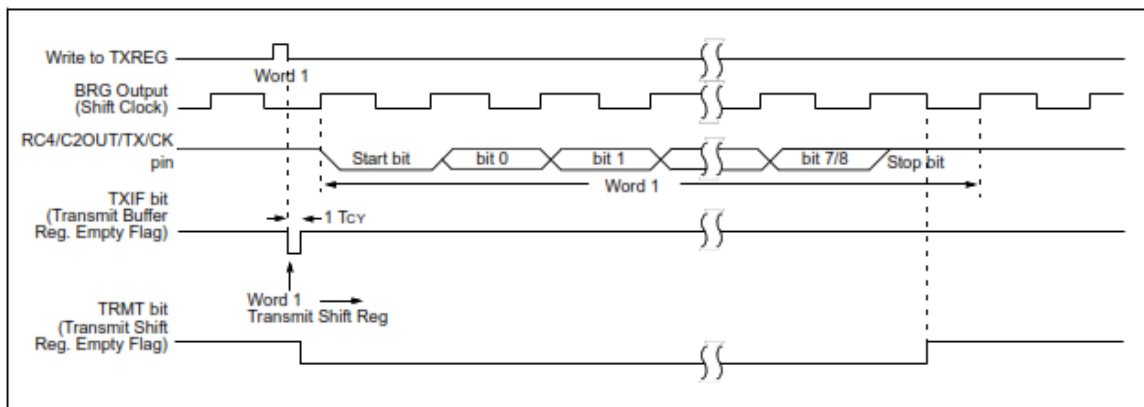
The TRMT bit of the TXSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is

transferred to the TSR register from the TXREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

**Note:** The TSR register is not mapped in datamemory, so it is not available to the user.

#### Asynchronous Transmission Set-up:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the 8 Least Significant data bits are an address when the receiver is set for address detection.
4. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
5. If interrupts are desired, set the TXIE interrupt enable bit. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
6. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
7. Load 8-bit data into the TXREG register. This will start the transmission.



**Figura A.10.** Asynchronous transmission

**Tabla A.11***Registros asociados con la transmisión asíncrona*

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	TOIE	INTE	RAIE	TOIF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCREG	EUSART Receive Data Register								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

**Legend:** x = unknown, — = unimplemented read as '0'. Shaded cells are not used for Asynchronous Transmission.

### A.3.2.3. RECEPCION ASINCRONA DEL EUSART

The Asynchronous mode would typically be used in RS-232 systems. The receiver block diagram is shown in Figure 10-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all 8 or 9 bits of the character have been shifted in, they are immediately transferred to a two character First-InFirst-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third carácter before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accesible by software. Access to the received data is via the RCREG register.

#### Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state. Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the RX/DT I/O pin as an input. If the RX/DT pin is shared with an analog peripheral the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

**Note:** When the SPEN bit is set the TX/CK I/O pin is automatically configured as an output, regardless of the state of the corresponding TRIS bit and whether or not the EUSART transmitter is enabled. The PORT latch is disconnected from the output driver so it is not possible to use the TX/CK pin as a general purpose output.

## Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

**Note:** If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared.

## Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software. RCIF interrupts are enabled by setting the following bits:

- RCIE interrupt enable bit of the PIE1 register
- PEIE peripheral interrupt enable bit of the INTCON register
- 
- GIE global interrupt enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

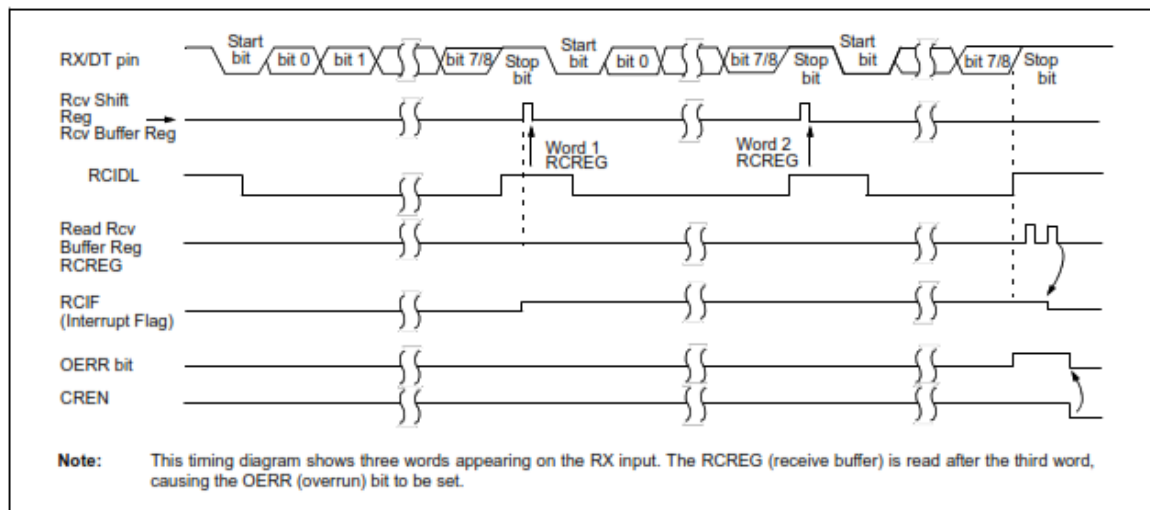
## Asynchronous Reception Set-up:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 10.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
3. If interrupts are desired, set the RCIE interrupt enable bit and set the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Enable reception by setting the CREN bit.

6. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.

7. Read the RCSR register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit. 8. Get the received 8 Least Significant data bits from the receive buffer by reading the RCRREG register.

9. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.



**Figura A.11.** Asynchronous reception

**Tabla A.12**

Registers associated with asynchronous reception

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	TOIE	INTE	RAIE	TOIF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
RCRREG	EUSART Receive Data Register								0000 0000	0000 0000
RCSR	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
TXRREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

**Legend:** x = unknown, — = unimplemented read as '0'. Shaded cells are not used for Asynchronous Reception.

#### A.3.2.4. EL GENERADOR DE BAUD RATE (BRG) DEL EUSART

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCTL register selects 16-bit mode.

The SPBRGH, SPBRG register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCTL register. In Synchronous mode, the BRGH bit is ignored.

Table A.13 contains the formulas for determining the baud rate. Figure A.12 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table A.13. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRG register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

For a device with  $F_{OSC}$  of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \frac{F_{OSC}}{64([SPBRGH:SPBRG] + 1)}$$

Solving for SPBRGH:SPBRG:

$$X = \frac{\frac{F_{OSC}}{\text{Desired Baud Rate}} - 1}{64}$$
$$= \frac{\frac{16000000}{9600} - 1}{64}$$
$$= [25.042] = 25$$
$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$
$$= 9615$$
$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$
$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

**Figure A.12.** Calculating baud rate error.

**Tabla A.13***Baud rate formulas*

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{osc}/[64 (n+1)]$
0	0	1	8-bit/Asynchronous	$F_{osc}/[16 (n+1)]$
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	$F_{osc}/[4 (n+1)]$
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

**Legend:** x = Don't care, n = value of SPBRGH, SPBRG register pair

**Tabla A.14***Registers associated with the baud rate generator*

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

**Legend:** x = unknown, — = unimplemented read as '0'. Shaded cells are not used for the Baud Rate Generator.

### A.3.3. MODULO TIMER1

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Optional LP oscillator
- Synchronous or asynchronous operation
- Timer1 gate (count enable) via comparator or T1G pin
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)

#### A.3.3.1. OPERACIÓN DEL TIMER1

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer. When used with an external clock source, the module can be used as either a timer or counter.

### A.3.3.2. SELECCIÓN DE FUENTE DE RELOJ

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is FOSC/4. When TMR1CS = 1, the clock source is supplied externally.

Clock Source	TMR1CS	Clock Source
FOSC/4	0	FOSC/4
T1CKI pin	1	T1CKI pin

### A.3.3.3. PRESCALER DEL TIMER1

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

### A.3.3.4. OSCILADOR DEL TIMER1

A low-power 32.768 kHz crystal oscillator is built-in between pins OSC1 (input) and OSC2 (amplifier output). The oscillator is enabled by setting the T1OSCEN control bit of the T1CON register. The oscillator will continue to run during Sleep.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator or when in LP oscillator mode. The user must provide a software time delay to ensure proper oscillator start-up.

TRISA5 and TRISA4 bits are set when the Timer1 oscillator is enabled. RA5 and RA4 bits read as '0' and TRISA5 and TRISA4 bits read as '1'.

**Note:** The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1.

### A3.3.5 REGISTRO DE CONTROL DEL TIMER1

**Tabla A.15**

T1CON: Registro del control del timer1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV <sup>(1)</sup>	TMR1GE <sup>(2)</sup>	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown



- bit 7 **T1GINV**: Timer1 Gate Invert bit<sup>(1)</sup>  
 1 = Timer1 gate is active high (Timer1 counts when gate is high)  
 0 = Timer1 gate is active low (Timer1 counts when gate is low)
- bit 6 **TMR1GE**: Timer1 Gate Enable bit<sup>(2)</sup>  
 If TMR1ON = 0:  
 This bit is ignored  
 If TMR1ON = 1:  
 1 = Timer1 is on if Timer1 gate is active  
 0 = Timer1 is on
- bit 5-4 **T1CKPS<1:0>**: Timer1 Input Clock Prescale Select bits  
 11 = 1:8 Prescale Value  
 10 = 1:4 Prescale Value  
 01 = 1:2 Prescale Value  
 00 = 1:1 Prescale Value
- bit 3 **T1OSCEN**: LP Oscillator Enable Control bit  
 If INTOSC without CLKOUT oscillator is active:  
 1 = LP oscillator is enabled for Timer1 clock  
 0 = LP oscillator is off  
 Else:  
 This bit is ignored. LP oscillator is disabled.
- bit 2 **T1SYNC**: Timer1 External Clock Input Synchronization Control bit  
 TMR1CS = 1:  
 1 = Do not synchronize external clock input  
 0 = Synchronize external clock input  
 TMR1CS = 0:  
 .....This bit is ignored. Timer1 uses the internal clock
- bit 1.....**TMR1CS**: Timer1 Clock Source Select bit  
 1 = External clock from T1CKI pin (on the rising edge)  
 0 = Internal clock (FOSC/4)
- bit 0 **TMR1ON**: Timer1 On bit  
 1 = Enables Timer1  
 0 = Stops Timer1

**Note 1:** T1GINV bit inverts the Timer1 gate logic, regardless of source.

**2:** TMR1GE bit must be set to use either T1G pin or C2OUT, as selected by the T1GSS bit of the CM2CON1 register, as a Timer1 gate source.

**Table A.16**

*Resumen de registros asociados con timer1*

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	00-- --10
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	EEIE	ADIE	RCIE	C2IE	C1IE	OSFIE	TXIE	TMR1IE	0000 0000	0000 0000
PIR1	EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF	0000 0000	0000 0000
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuu uuuu

**Legend:** x = unknown, u = unchanged, – = unimplemented, read as ‘0’. Shaded cells are not used by the Timer1 module.

## A.4. MODULO GSM/GPRS SIM900

El shield como tal no tiene un data sheet y no se encontró un manual de usuario para este producto. Al momento de adquirir este producto no incluía un manual de usuario. De hecho ningún proveedor local o extranjero tiene el manual de usuario y la información a cerca de este shield es escasa dado que es un shield que solo debe insertarse en la placa Arduino UNO y está listo para operar. Aunque no se tiene un data sheet para poder describir este shield, se puede describir el chip SIM900 el cual es el corazón del módulo GSM/GPRS, que si tiene un data sheet.

### A.4.1 DESCRIPCION GENERAL DEL SIM900

Designed for global market, SIM900 is a quad-band GSM/GPRS module that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM900 features GPRS multi-slot class 10/ class 8 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

With a tiny configuration of 24\*24\*3mm, SIM900 can meet almost all the space requirements in user applications, such as M2M, smart phone, PDA and other mobile devices.

SIM900 has 68 SMT pads, and provides all hardware interfaces between the module and customers' boards.

- Serial port and debug port can help user easily develop user's applications.
- Audio channel which includes a microphone input and a receiver output.
- Programmable general purpose input and output.
- The keypad and SPI display interfaces will give users the flexibility to develop customized applications.

SIM900 is designed with power saving technique so that the current consumption is as low as 1.0mA in sleep mode.

SIM900 integrates TCP/IP protocol and extended TCP/IP AT commands which are very useful for data transfer applications.

#### A.4.1.1. CARACTERISTICAS CLAVE DEL SIM900

**Tabla A.17**

*Características clave del SIM900*

Feature	Implementation
Power saving	Typical power consumption in sleep mode is 1.0mA ( BS-PA-MFRMS=9 )

Frequency bands	<ul style="list-style-type: none"> <li>• SIM900 Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. SIM900 can search the 4 frequency bands automatically. The frequency bands also can be set by AT command “AT+CBAND”.</li> <li>• Compliant to GSM Phase 2/2+</li> </ul>
Transmitting power	<ul style="list-style-type: none"> <li>• Class 4 (2W) at GSM 850 and EGSM 900</li> <li>• Class 1 (1W) at DCS 1800 and PCS 1900</li> </ul>
GPRS connectivity	<ul style="list-style-type: none"> <li>• GPRS multi-slot class 10 (default)</li> <li>• GPRS multi-slot class 8 (option)</li> </ul>
Temperature range	<ul style="list-style-type: none"> <li>• Normal operation: -30°C ~ +80°C</li> <li>• Restricted operation: -40°C ~ -30°C and +80 °C ~ +85°C*</li> <li>• Storage temperature -45°C ~ +90°C</li> </ul>
Data GPRS	<ul style="list-style-type: none"> <li>• GPRS data downlink transfer: max. 85.6 kbps</li> <li>• GPRS data uplink transfer: max. 42.8 kbps</li> <li>• Coding scheme: CS-1, CS-2, CS-3 and CS-4</li> <li>• Integrate the TCP/IP protocol.</li> <li>• Support Packet Broadcast Control Channel (PBCCH)</li> </ul>
CSD	Support CSD transmission
USSD	Unstructured Supplementary Services Data (USSD) support
SMS	<ul style="list-style-type: none"> <li>• MT, MO, CB, Text and PDU mode</li> <li>• SMS storage: SIM card</li> </ul>
FAX	Group 3 Class 1
SIM interface	Support SIM card: 1.8V, 3V
External antenna	Antenna pad
Audio features	Speech codec modes: <ul style="list-style-type: none"> <li>• Half Rate (ETS 06.20)</li> <li>• Full Rate (ETS 06.10)</li> <li>• Enhanced Full Rate (ETS 06.50 / 06.60 / 06.80)</li> <li>• Adaptive multi rate (AMR)</li> <li>• Echo Cancellation</li> <li>• Noise Suppression</li> </ul>
Serial port and debug port	<b>Serial port:</b> <ul style="list-style-type: none"> <li>• Full modem interface with status and control lines, unbalanced, asynchronous.</li> <li>• 1200bps to 115200bps.</li> <li>• Can be used for AT commands or data stream.</li> </ul>

	<ul style="list-style-type: none"> <li>Support RTS/CTS hardware handshake and software ON/OFF flow control.</li> <li>Multiplex ability according to GSM 07.10 Multiplexer Protocol.</li> <li>Autobauding supports baud rate from 1200 bps to 57600bps.</li> </ul> <p><b>Debug port:</b></p> <ul style="list-style-type: none"> <li>Null modem interface DBG_TXD and DBG_RXD.</li> <li>Can be used for debugging and upgrading firmware.</li> </ul>
Phonebook management	Support phonebook types: SM, FD, LD, RC, ON, MC.
SIM application toolkit	GSM 11.14 Release 99
Real time clock	Support RTC
Physical characteristics	Size: 24*24*3mm Weight: 3.4g
Firmware upgrad	Firmware upgradeable by debug port.

\*  
\*SIM900 does work at this temperature, but some radio frequency characteristics may deviate from the GSM specification.

**Tabla A.18**

*Códigos de esquema y máxima velocidad de datos en la red sobre el aire*

Coding scheme	1 timeslot	2 timeslot	4 timeslot
CS-1	9.05kbps	18.1kbps	36.2kbps
CS-2	13.4kbps	26.8kbps	53.6kbps
CS-3	15.6kbps	31.2kbps	62.4kbps
CS-4	21.4kbps	42.8kbps	85.6kbps

#### A.4.1.2. MODOS DE OPERACION

**Tabla A.19**

*Descripción de los modos de operacion*

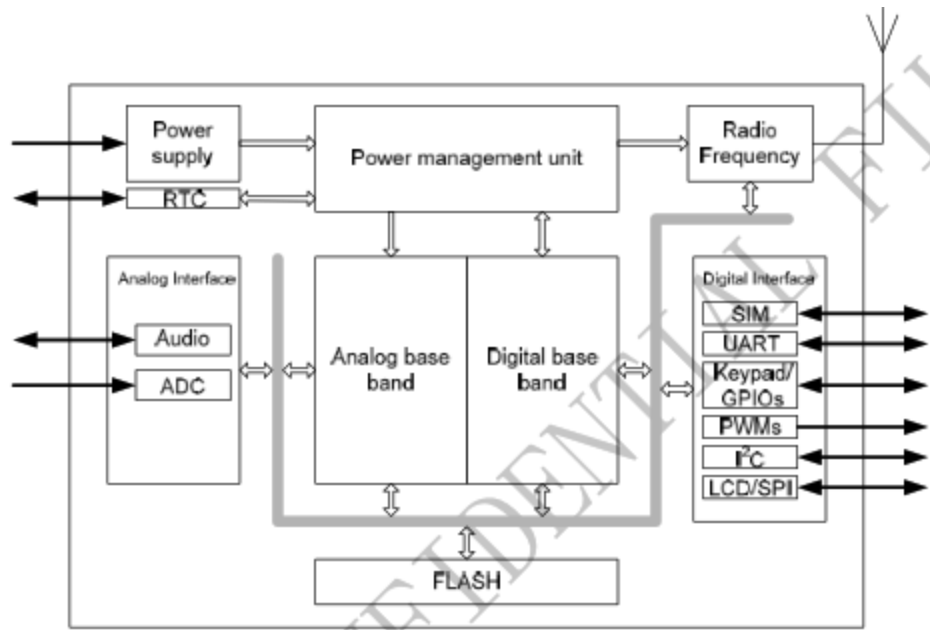
Mode	Function	
Normal operation	GSM/GPRS SLEEP	Module will automatically go into sleep mode if the conditions of sleep mode are enabling and there is no on air and no hardware interrupt (such as GPIO interrupt or data on serial port). In this case, the current consumption of module will reduce to the minimal level. In sleep mode, the module can still receive paging message and SMS.
	GSM IDLE	Software is active. Module registered to the GSM network, and the module is ready to communicate.
	GSM TALK	Connection between two subscribers is in progress. In this case, the power consumption depends on network

		settings such as DTX off/on, FR/EFR/HR, hopping sequences, antenna.
	GPRS STANDBY	Module is ready for GPRS data transfer, but no data is currently sent or received. In this case, power consumption depends on network settings and GPRS configuration.
	GPRS DATA	There is GPRS data transfer (PPP or TCP or UDP) in progress. In this case, power consumption is related with network settings (e.g. power control level); uplink/downlink data rates and GPRS configuration (e.g. used multi-slot settings).
Power down	Normal power down by sending the AT command “AT+CPOWD=1” or using the PWRKEY. The power management unit shuts down the power supply for the baseband part of the module, and only the power supply for the RTC is remained. Software is not active. The serial port is not accessible. Power supply (connected to VBAT) remains applied.	
Minimum functionality mode	AT command “AT+CFUN” can be used to set the module to a minimum functionality mode without removing the power supply. In this mode, the RF part of the module will not work or the SIM card will not be accessible, or both RF part and SIM card will be closed, and the serial port is still accessible. The power consumption in this mode is lower than normal mode.	

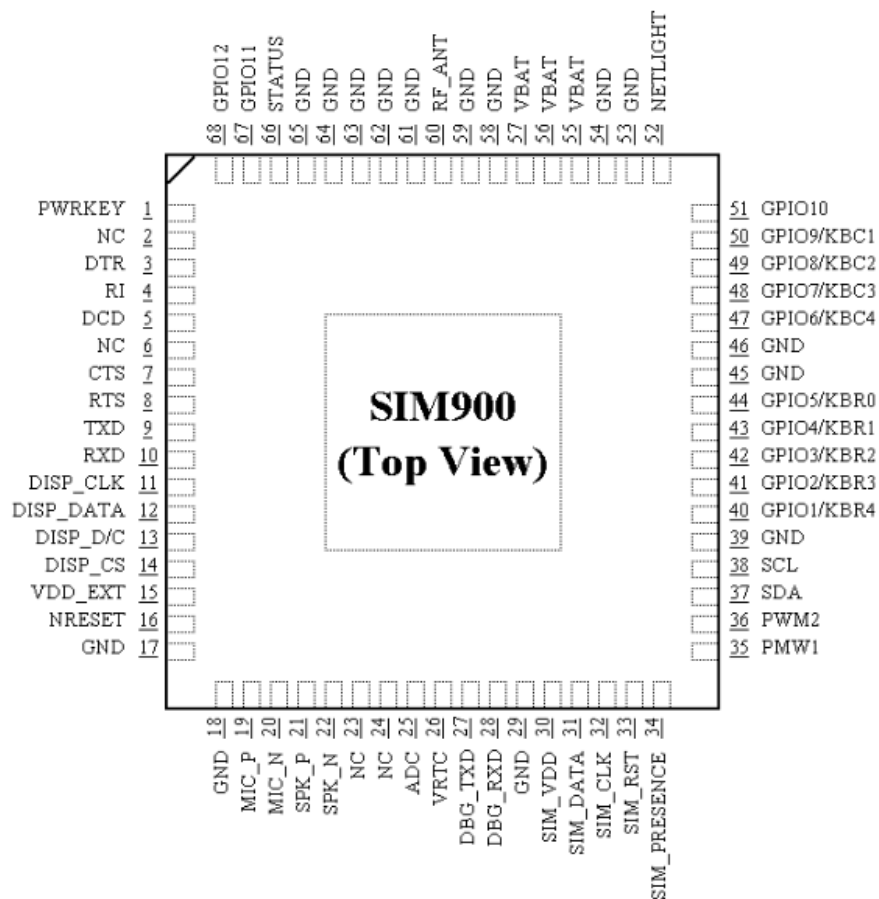
#### A.4.1.3. DIAGRAMA FUNCIONAL DEL SIM900

The following figure shows a functional diagram of SIM900:

- The GSM baseband engine
- Flash
- The GSM radio frequency part
- The antenna interface
- The other interfaces



**Figura A.13.** Diagrama funcional del SIM900



**Figura A.14.** Diagrama de pines

#### A.4.1.4. DESCRIPCION DE PINES.

**Tabla A.20.**

Descripción de pines del SIM900

Pin name	Pin number	I/O	Description	Comment
Power supply				
VBAT	55,56,57	I	Power supply	
VRTC	26	I/O	Power supply for RTC	It is recommended to connect with a battery or a capacitor (e.g. 4.7uF).
VDD_EXT	15	O	2.8V output power supply	If it is unused, keep open.
GND	17,18,29,39,45,46,53,54,58,59,61,62,63,64,65		Ground	
Power on/down				
PWRKEY	1	I	PWRKEY should be pulled low at least 1 second and then released to power on/down the module.	Pulled up internally.
Audio interfaces				
MIC_P	19	I	Differential audio input	If these pins are unused, keep open.
MIC_N	20			
SPK_P	21	O	Differential audio output	
SPK_N	22			
Status				
STATUS	66	O	Power on status	If these pins are unused,keep open.
NETLIGHT	52	O	Network status	
LCD interface				
DISP_CLK	11	O	Display interface	If these pins are unused, keep open.
DISP_DATA	12	I/O		
DISP_D/C	13	O		
DISP_CS	14	O		
Keypad interface / GPIOs				
GPIO5/KBR0	44	I/O	GPIO5/keypad row 0	If these pins are unused, keep open.
GPIO4/KBR1	43		GPIO4/keypad row 1	
GPIO3/KBR2	42		GPIO3/keypad row 2	
GPIO2/KBR3	41	I/O	GPIO2/keypad row 3	If these pins are unused,

GPIO1/KBR4	40		GPIO1/keypad row 4	keep open.
GPIO9/KBC1	50		GPIO9/keypad column 1	
GPIO8/KBC2	49		GPIO8/keypad column 2	
GPIO7/KBC3	48		GPIO7/keypad column 3	
GPIO6/KBC4	47		GPIO6/keypad column 4	
GPIO10	51		GPIO10	
GPIO11	67		GPIO11	
GPIO12	68		GPIO12	
Serial port				
RXD	10	I	Receive data	This pin should be pulled up to VDD_EXT externally.  If these pins are unused, keep open.
TXD	9	O	Transmit data	
RTS	8	I	Request to send	
CTS	7	O	Clear to send	
DCD	5	O	Data carrier detect	
RI	4	O	Ring indicator	
DTR	3	I	Data terminal ready	
Debug interface				
DBG_TXD	27	O	For debugging and upgrading firmware	If these pins are unused, keep open.
DBG_RXD	28	I		
SIM interface				
SIM_VDD	30	O	Voltage supply for SIM card. Support 1.8V or 3V SIM card	All signals of SIM interface should be protected against ESD with a TVS diode array.
SIM_DATA	31	I/O	SIM data input/output	
SIM_CLK	32	O	SIM clock	
SIM_RST	33	O	SIM reset	
SIM_PRESENCE	34	I	SIM card detection	If it is unused, keep open.
ADC				
ADC	25	I	Input voltage range: 0V ~ 2.8V	If it is unused, keep open.



External reset				
NRESET	16	I	Reset input(Active low)	Recommend connecting a 100nF capacitor.
Pulse width modulation( PWM )				
PWM1	35	O	PWM	If these pins are unused, keep open.
PWM2	36	O	PWM	
RF interface				
RF_ANT	60	I/O	Radio antenna connection	Impedence must be controlled to 50Ω.
Not connect				
NC	2,6,23,24	-		These pins should be kept open.

## ANEXO B: COMANDOS AT DEL SIM900 VERSION 1.07

El documento oficial proveído por SIMCOM contiene una explicación detallada de todos los comandos AT, en su versión 1.07, soportados por el SIM900. En este anexo se mostraran aquellos comandos AT que están relacionados con la transmisión de datos via GPRS, como son los comandos AT para la aplicación IP y la aplicación HTTP.

### B.1 COMANDOS AT PARA LA APLICACIÓN IP

**Tabla B.1**

*Descripcion general*

Command	Description
AT+SAPBR	BEARER SETTINGS FOR APPLICATIONS BASED ON IP

**Tabla B.2**

*Descripcion detallada de los comandos*

AT+SAPBR Bearer Settings for Applications Based on IP	
Test Command AT+SAPBR=?	Response +SAPBR: (0-5),(1-3), "ConParamTag","ConParamValue"  OK
Write Command AT+SAPBR=<cmd_type>,<cid>[,<ConParamTag>,<ConParamValue>]	Response OK  If <cmd_type>=2 +SAPBR: <cid>,<Status>,<IP_Addr> OK If <cmd_type>=4 +SAPBR: <ConParamTag>,<ConParamValue> OK
	Unsolicited Result Code +SAPBR <cid>: DEACT
	Parameters <cmd_type> 0 Close bearer 1 Open bearer 2 Query bearer

	3 Set bearer parameters 4 Get bearer parameters 5 Save the values of parameters to NVRAM <cid> Bearer profile identifier <Status> 0 Bearer is connecting 1 Bearer is connected 2 Bearer is closing 3 Bearer is closed <ConParamTag> Bearer parameter "CTYPE" Type of Internet connection. Value refer to <ConParamValue_ConType> "APN" Access point name string: maximum 50 characters "USER" User name string: maximum 50 characters "PWD" Password string: maximum 50 characters "PHONENUM" Phone number for CSD call "RATE" CSD connection rate. For value refer to <ConParamValue_Rate> <ConParamValue> Bearer parameter value <ConParamValue_ConType> "CSD" Circuit-switched data call. "GPRS" GPRS connection. <ConParamValue_Rate> 0 2400 1 4800 2 9600 3 14400 <IP_Addr> The IP address of bearer
Reference	Note This command is applied to activate some applications such as HTTP, FTP.

## B.2. COMANDOS AT PARA LA APLICACION HTTP

SIM900 has an embedded TCP/IP stack that is driven by AT commands and enables the host application to easily access the Internet HTTP service. This chapter is a reference guide to all the AT commands and responses defined to use with the TCP/IP stack in HTTP Service.

**Tabla B.3**

*Descripción general*

Command	Description
AT+HTTPINIT	INITIALIZE HTTP SERVICE
AT+HTTPTERM	TERMINATE HTTP SERVICE

AT+HTTTPARA	SET HTTP PARAMETERS VALUE
AT+HTTPDATA	INPUT HTTP DATA
AT+HTTPACTION	HTTP METHOD ACTION
AT+HTTPREAD	READ THE HTTP SERVER RESPONSE
AT+HTTPSCONT	SAVE HTTP APPLICATION CONTEXT
AT+HTTPSTATUS	READ HTTP STATUS

**Tabla B.4**

*Descripción detalla de los comandos*

<b>AT+HTTPINIT Initialize HTTP Service</b>	
Test Command <b>AT+HTTPINIT=?</b>	Response <b>OK</b>
Execution Command <b>AT+HTTPINIT</b>	Response <b>OK</b>  If error is related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
Reference	Note HTTPINIT should first be executed to initialize the HTTP service.
<b>AT+HTTPTERM Terminate HTTP Service</b>	
Test Command <b>AT+HTTPTERM M=?</b>	Response <b>OK</b>
Execution command <b>AT+HTTPTERM</b>	Response <b>OK</b>  If error is related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
<b>AT+HTTTPARA Set HTTP Parameters Value</b>	
Test Command <b>AT+HTTTPARA =?</b>	Response <b>+HTTTPARA: "HTTTPParamTag","HTTTPParamValue"</b>  <b>OK</b>
Read Command <b>AT+HTTTPARA ?</b>	Response <b>+HTTTPARA: &lt;HTTTPParamTag&gt;,&lt;HTTTPParamValue&gt;</b>  <b>OK</b>
Write Command <b>AT+HTTTPARA =&lt;HTTTPParamT</b>	Response <b>OK</b>  If error is related to ME functionality:

<b>ag&gt;,&lt;HTTPParamValue&gt;[&lt;UserDataDelimiter&gt;]</b>	<b>+CME ERROR: &lt;err&gt;</b>  Parameters <b>&lt;HTTPParamTag&gt;</b> HTTP Parameter "CID" (Mandatory Parameter) Bearer profile identifier "URL" (Mandatory Parameter) HTTP client URL, the maximum length is 500 bytes. "http://server'/path':tcpPort' " "server": FQDN or IP-address "path": path of file or directory "tcpPort": default value is 80. "UA" Refer to "IETF-RFC 2616". The user agent string which is set by the application to identify the mobile. Usually this parameter is set as operation system and software version information. The maximum length is 200 bytes. "PROIP" Default value is "SIMCOM_MODULE". "PROPORT" The IP address of HTTP proxy server "REDIR" The port of HTTP proxy server This flag controls the redirection mechanism of the SIM900 when it is acting as HTTP client (numeric). If the server sends a redirect code (range 30x), the client will automatically send a new HTTP request when the flag is set to (1). "BREAK" Default value is 0 (no redirection). "BREAKEND" Parameter for HTTP method "GET", used for resuming broken transfer. Parameter for HTTP method "GET", used for resuming broken transfer. which is used together with "BREAK", If the value of "BREAKEND" is bigger than "BREAK", the transfer scope is from "BREAK" to "BREAKEND". If the value of "BREAKEND" is smaller tan "BREAK", the transfer scope is from "BREAK" to the end of the file. If both "BREAKEND" and "BREAK" are 0, the resume broken transfer function is disabled. HTTP session timeout value, scope: 30-1000 second. Default value is 120 seconds. "TIMEOUT" HTTP Parameter value. Type and supported content depend on related <HTTPParamTag>. "CONTENT" Used to set the “Content-Type” field in HTTP header "USERDATA" Used to set the user’s data in HTTP header <b>&lt;HTTPParamValue&gt;</b> HTTP Parameter value. Type and supported content depend on related <HTTPParamTag>.
---	---

	<b>&lt;UserdataDelimiter&gt;</b> The delimiter in the string of "USERDATA" will be replaced by 0x0D0x0A .
Reference	Note Not all the HTTP Server supports "BREAK" and "BREAKEND" parameters
<b>AT+HTTPDATA Input HTTP Data</b>	
Test Command <b>AT+HTTPDATA=?</b>	Response <b>+HTTPDATA:</b> (list of supported <b>&lt;size&gt;s</b> ),(list of supported <b>&lt;time&gt;s</b> )  <b>OK</b>
Write Command <b>AT+HTTPDATA=&lt;size&gt;,&lt;time&gt;</b>	Response <b>DOWNLOAD</b>  <b>OK</b> If error is related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
	Parameters <b>&lt;size&gt;</b> Size in bytes of the data to POST. 1-102400 or 1-318976 (bytes) the maximum size depends on the module. 0 means delete all the content. <b>&lt;time&gt;</b> 1000-120000 (millisecond) Maximum time in milliseconds to input data.
Reference	Note It is strongly recommended to set enough time to input all data with the length of <b>&lt;size&gt;</b> .
<b>AT+HTTPACTION HTTP Method Action</b>	
Test Command <b>AT+HTTPACTION=?</b>	Response <b>+HTTPACTION: (0-2)</b>  <b>OK</b>
Write Command <b>AT+HTTPACTION=&lt;Method&gt;</b>	Response <b>OK</b> If error is related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
	Unsolicited Result Code <b>+HTTPACTION: &lt;Method&gt;,&lt;StatusCode&gt;,&lt;DataLen&gt;</b>
	Parameters <b>&lt;Method&gt;</b> HTTP method specification: 0 GET 1 POST 2 HEAD <b>&lt;StatusCode&gt;</b> HTTP Status Code responded by remote server, it identifier refer to HTTP1.1(RFC2616)

	100 Continue 101 Switching Protocols 200 OK 201 Created 202 Accepted 203 Non-Authoritative Information 204 No Content 205 Reset Content 206 Partial Content 300 Multiple Choices 301 Moved Permanently 302 Found 303 See Other 304 Not Modified 305 Use Proxy 307 Temporary Redirect 400 Bad Request 401 Unauthorized 402 Payment Required 403 Forbidden 404 Not Found 405 Method Not Allowed 406 Not Acceptable 407 Proxy Authentication Required 408 Request Time-out 409 Conflict 410 Gone 411 Length Required 412 Precondition Failed 413 Request Entity Too Large 414 Request-URI Too Large 415 Unsupported Media Type 416 Requested range not satisfiable 417 Expectation Failed 500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Time-out 505 HTTP Version not supported 600 Not HTTP PDU 601 Network Error 602 No memory 603 DNS Error 604 Stack Busy  <DataLen> the length of data got
<b>AT+HTTPREAD Read the HTTP Server Response</b>	
Test Command <b>AT+HTTPREA</b>	Response <b>+HTTPREAD:</b> (list of supported <start_address>s),(list of supported<

<b>D=?</b>	<b>byte_size&gt;s)</b>  <b>OK</b>
Write Command <b>AT+HTTPREAD=&lt;start_addresses&gt;,&lt;byte_size&gt;</b>	Response <b>+HTTPREAD: &lt;data_len&gt;</b> <b>&lt;data&gt;</b>  <b>OK</b>  Read data when AT+HTTPACTION=0 or AT+HTTPDATA is executed.  If <byte_size> is bigger than the data size received, module will only return actual data size.  If error is related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
	Parameters <b>&lt;data&gt;</b> Data from HTTP server or user input. <b>&lt;start_address&gt;</b> The starting point for data output. 0-318976 or 0-102400 (bytes), the max value is due to the module used. <b>&lt;byte_size&gt;</b> The length for data output. 1-318976 or 1-102400 (bytes), the max value is due to the module used. <b>&lt;data_len&gt;</b> The actual length for data output.
Execution Command <b>AT+HTTPREAD</b>	Response <b>+HTTPREAD: &lt;data_len&gt;</b> <b>&lt;data&gt;</b>  <b>OK</b>  Read all data when AT+HTTPACTION=0 or AT+HTTPDATA is executed.  If error is related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
<b>AT+HTTPSCONT Save HTTP Application Context</b>	
Read Command <b>AT+HTTPSCONT?</b>	Response TA returns HTTP Application Context, which consists of the following AT Command parameters. <b>+HTTPSCONT: &lt;mode&gt;</b> <b>CID: &lt;value&gt;</b> <b>URL: &lt;value&gt;</b> <b>UA: &lt;value&gt;</b> <b>PROIP: &lt;value&gt;</b> <b>PROPORT: &lt;value&gt;</b> <b>REDIR: &lt;value&gt;</b> <b>BREAK: &lt;value&gt;</b>



	<b>BREAKEND: &lt;value&gt;</b>  <b>OK</b>
	Parameters <b>&lt;mode&gt;</b> 0 Saved, the value from NVRAM 1 Unsaved, the value from RAM
Execution Command <b>AT+HTTPSCONT</b>	Response TA saves HTTP Application Context which consists of following AT Command parameters, and when system is rebooted, the parameters will be loaded automatically. <b>OK</b>  If error is related to ME functionality: <b>+CME ERROR: &lt;err&gt;</b>
<b>AT+HTTPSTATUS Read HTTP Status</b>	
Test Command <b>AT+HTTPSTAT US=?</b>	Response <b>OK</b>
Read Command <b>AT+HTTPSTAT US?</b>	Response <b>+HTTPSTATUS: &lt;mode&gt;,&lt;status&gt;,&lt;finish&gt;,&lt;remain&gt;</b>  <b>OK</b>
	Parameter: <b>&lt;mode&gt;</b> GET POST HEAD <b>&lt;status&gt;</b> 0 idle 1 receiving 2 sending <b>&lt;finish&gt;</b> The amount of data which have been transmitted. <b>&lt;remain&gt;</b> The amount of data remaining to be sent or received.

## ANEXO C: DIRECTIVAS Y FUNCIONES PREDEFINIDAS EN CCS

Las directivas de este anexo están referidas a las utilizadas en el sistema que tienen que ver con la configuración del módulo USART y del timer1 del microcontrolador PIC. La lista de funciones han sido tomadas del García, 2008.

### C.1. PARAMETROS DE CONFIGURACIÓN DEL USART

En C la configuración genérica del USART es: `#USE(parámetro)`. A través de esta directiva se pueden configurar varios parámetros. Antes de utilizarse se debe definir la directiva `#USE DELAY`.

**Tabla C.1**

*Descripción de los parámetros soportados por #USE()*

Parámetro	Descripción
BAUD=X	Pin de transmisión
RCV=pin	Pin de recepción
FORCE_SW	Usa un software UART en lugar del hardware aun cuando se especifican los pines del hardware.
RESTART_WDT	Hace que la función GETC() ponga a cero el WDT mientras espera un carácter.
BRGH1OK	Permite velocidades de transmisión bajas en chips que tienen problemas de transmisión.
ENABLE=pin	El pin especificado estará a nivel alto durante la transmisión. Utilizando en transmisión 485.
DEBUGGER	Permite depuración a través del ICD. Usar XMIT y RCV para cambiar el pin (debe ser el mismo en ambos)
RESTART_WDT	Provoca que la función GETC() borre el WDT si espera un carácter.
INVERT	Invierte la polaridad de los pines serie (normalmente no es necesario con el convertidor de nivel, como el MAX232). No puede usarse con el USART interno.
PARITY=X	Donde X es N, E u O
BITS=X	DONDE X ES 5-9 (no puede usarse 5-7 con el USART interno)
FLOAT_HIGH	Se utiliza para las salidas de colector abierto.

ERRORS	Indica al compilador que guarde los errores recibidos en la variable RS232_ERRORS para restablecerlos cuando se producen.
SAMPLE_EARLY	No se puede utilizar con USART interno. Provoca que el muestreo del dato a través de la función GETC() se realice al principio de un bit de tiempo.
RETURN=pin	Para FLOAT_HIGH y MULTI_MASTER, este pin se usa para leer la señal de retorno. Por defecto, para FLOAT_HIGH es XMIT y para MULTI_MASTER ES RCV.
MULTI_MASTER	Usa el pin de RETURN para determinar si otro master en el bus está transmitiendo al mismo tiempo. Si se detecta una colisión, el bit 6 se pone a 1 en RS232_ERRORS y todos
LONG_DATA	Permite manejar INT16 a las funciones GETC() y PUTC(). En formatos de datos de 9 bits.
DISABLE_INTS	Provoca la des habilitación de interrupciones cuando se ejecuta GETC() y PUTC() evitando distorsiones en los datos.
STOP=x	Numero de bits de stop (por defecto 1)
TIMEOUT=x	Para establecer el tiempo que GETC() espera un carácter (en ms). Si no se recibe carácter en este tiempo, el RS232_ERRORS se pone a 0.
SYNC_SLAVE	Provoca una línea RS232 en modo esclavo síncrono, haciendo la patilla de recepción como entrada de reloj y la de transmisión como entrada/salida de datos.
SYNC_MASTER	Provoca una línea RS232 en modo maestro síncrono, haciendo la patilla de recepción como salida de reloj y la de transmisión como entrada/salida de datos.
SYNC_MATER_CONT	Provoca una línea RS232 en modo maestro síncrono en modo continuo, haciendo la patilla de recepción como salida de reloj y la de transmisión como entrada/salida de datos.
UART1	Configurar el XMIT y RCV para el USART1
UART2	Configurar el XMIT y RCV para el USART2

## C.2 FUNCIONES PREDEFINIDAS ASOCIADAS AL USART

**Tabla C.2**

*Funciones predefinidas*

Función	
setup_uart(baud)	baud es una constante que define la velocidad.

set_uart_speed(baud)	
putc(cdata)	cdata es un carácter de 8 bits. Esta función envía un carácter mediante la patilla XMIT. La directiva #USE RS-232 debe situarse siempre antes de utilizar esta función.
putchar(cdata)	
puts(string)	string: cadena de caracteres o matriz de caracteres terminada con un 0. La función puts() manda los caracteres de la cadena, uno a uno, a través del bus RS-232 utilizando la función PUTC(); detrás de la cadena envía un RETURN (13) y un retorno de carro (10).
printf(fname, cstring, values...)	<b>cstring</b> es una cadena de caracteres (constante) o matriz de caracteres terminada con un 0. <b>fname:</b> las funciones a utilizar para escribir la cadena indicada; por defecto se utiliza al función PUTC(), que permite escribir en el bus RS-232. <b>values:</b> valores a incluir en la cadena separados por comas.
valor=getc()	value es un carácter de 8 bits. Espera recibir un carácter por la línea RS-232 y devuelve su valor. En los dispositivos con USART interno, se pueden almacenar hasta tres caracteres; para evitar esperas se puede usar la función kbhit()
valor=getch()	
valor=getchar()	
valor=kbhit()	valor es 0 (FALSE) si GETC() debe esperar un carácter; 1 (TRUE) si ya hay un carácter listo para ser leído por la función GETC().

### C.3. MODOS DE CONFIGURACION DEL TIMER1

**Tabla C.3**

*Modos de configuración del timer1*

Setup_Timer1(modo);
T1_INTERNAL
T1_EXTERNAL
T1_EXTERNAL_SYNC
T1_CLK_OUT
T1_DIV_BY_1
T1_DIV_BY_2
T1_DIV_BY_4
T1_DIV_BY_8

Donde modo está definido en el fichero de cabecera. Los distintos modos se pueden agrupar mediante el empleo de | (barra vertical).

## ANEXO D: CODIGOS DE PROGRAMACION

### D.1. CODIGO DEL MICROCONTROLADOR PIC16F688

```
/*Este programa en CCS permite leer un dato de nivel de agua, cada intervalo de tiempo,
y lo envía al módulo SIM900 indicándole el servidor en línea al cual debe ser subido*/
```

```
/*Programado por: Jorge Real*/
```

```
#include <16F688.h>
```

```
#include<string.h> //para usar puts() y strcmp()
```

```
#fuses INTRC_IO,NOWDT,NOPROTECT,NOBROWNOUT,NOPUT
```

```

#use delay(clock=4M)
#use standard_io(A)
#use standard_io(C)
#use rs232(baud=9600, xmit=pin_c4, rcv=pin_c5)

char letra=0x00;
char palabra[30]="";
int i;
char http[19]="+HTTPACTION:0,200,";
int result=0;
int bandera=0;
int bandera2=0;
char respuesta[10]="+CGATT: 1";
int band_resp=0;
signed int16 tiempo;
int16 calibrar[20];
int16 nivel_ref=0;
int cont=0;
int k=0;
int frec[19]={};
int max;
int k_max;
#define trig pin_C1
#define echo pin_C0

void parpadeo_A0(){
    output_high(PIN_A0);
    delay_ms(500);
    output_low(PIN_A0);
    delay_ms(500);
    output_high(PIN_A0);
    delay_ms(500);
    output_low(PIN_A0);
}

void parpadeo_A1(){
    output_high(PIN_A1);
    delay_ms(500);
    output_low(PIN_A1);
    delay_ms(500);
    output_high(PIN_A1);
    delay_ms(500);
    output_low(PIN_A1);
    delay_ms(500);
}

//Rutina de lectura de tiempo de recorrido del ultrasonido
void rutina_lectura_HC(){
    output_high(pin_C2);
    delay_us(20); //pulso de al menos 10us
    output_low(pin_C2);
    while(!input(pin_C3)){
    }
    set_timer1(0);
    while(input(pin_C3)){
    }
}

```

```

//Indicador visual de lectura de valor de referencia
void parpadeo_nivelref(){
    output_high(PIN_A0);
    delay_ms(300);
    output_low(PIN_A0);
    delay_ms(300);
    output_high(PIN_A0);
    delay_ms(300);
    output_low(PIN_A0);
}
/*Esta funcion encuentra un valor de referencia desde
donde medir el nivel de agua, el nivel cero del suelo*/
void obtenerNivelref(){
    for(cont=0;cont<20;cont++){
        rutina_lectura_HC();
        tiempo=get_timer1();
        calibrar[cont]=tiempo;
        parpadeo_nivelref();
    }
    max=0;
    for(k=0;k<19;k++){
        for(cont=K+1;cont<20;cont++){
            if(calibrar[k]==calibrar[cont])
                frec[k]++;
        }
        if(frec[k]>max){
            max=frec[k];
            k_max=k;
        }
    }
    nivel_ref=calibrar[k_max];
}
//Rutina de interrupcion por recepcion de datos
#int_RDA

void RDA_isr(){
    letra=0x00;
    if(kbhit()){
        letra=getc();
        if(letra=='r' || letra=='n' || letra=='0'){
            if(strcmp(respuesta,palabra)==0)
                band_resp=1;

            result=strspn(http,palabra);
            if(result==18){
                bandera=1;
            }
            else if(result==14){
                bandera2=1;
            }
            memset(palabra,0,sizeof(palabra));
            i=0;
            letra=0x00;
        }
        else
            {palabra[i]=letra;

```

```

    i++;
    letra=0x00;
}
}
}

void main () {
    output_low(PIN_A0); //Pin para LED
    output_low(PIN_A1); //Pin para LED
    output_low(PIN_A2); //Pin fijo en bajo
    output_low(PIN_A3); //Pin fijo en bajo
    output_low(PIN_A4); //pin fijo en bajo
    output_low(PIN_C1); //Pin encendido de SIM900
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);

    delay_ms(2000);

    etiqueta1:
    band_resp=0;
    while(input(PIN_C0)==0){ //Leer y verificar el estado del pin STS del SIM900
        parpadeo_A0();
        output_high( PIN_C1); //Pulso necesario para encender el sim900
        delay_ms(3000);
        output_low(PIN_C1);

        delay_ms(30000);
        parpadeo_A0();
    }

    while(nivel_ref==0)
        obtenerNivelref();

    while(band_resp==0){
        if(input(PIN_C0)==0)
            goto etiqueta1;
        printf("AT+CGATT?\r\n");
        parpadeo_A1();
    }

    //Comandos de configuracion de conexion a internet
    printf("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"\r\n");
    delay_ms(900);
    printf("AT+SAPBR=3,1,\"APN\",\"internet.movistar.ni\"\r\n");
    delay_ms(900);
    printf("AT+SAPBR=3,1,\"USER\",\"movistarni\"\r\n");
    delay_ms(900);
    printf("AT+SAPBR=3,1,\"PWD\",\"movistarni\"\r\n");
    delay_ms(900);
    printf("AT+SAPBR=1,1\r\n");
    delay_ms(9000);

    printf("AT+HTTPINIT\r\n");
    delay_ms(900);
    printf("AT+HTTTPARA=\"CID\",1\r\n");
    delay_ms(900);

```

```

while(1){
    if(input(PIN_C0)==0)
        goto etiqueta1;

    bandera=0;

    rutina_lectura_HC();
    tiempo=get_timer1();
    tiempo=nivel_ref-tiempo;//en microsegundos
    if(tiempo<0)
        tiempo=0;
    printf("AT+HTTTPARA=\"URL\", \"http://sensor-rt.000webhostapp.com/i.php?valor=\");
    delay_ms(50);
    printf("%Lu", tiempo);
    printf("\r\n");
    delay_ms(900);
    printf("AT+HTTPACTION=0\r\n"); //Sube dato con metodo GET
    delay_ms(8000);

    while(bandera==0){ //Mientras no se reciba un codigo de respuesta 200
        if(input(PIN_C0)==0)
            goto etiqueta1;

        if(bandera2==0){ //Si se reciba cualquier respuesta diferente 200
            output_high(PIN_A1);
            delay_ms(5000);
            output_low(PIN_A1);
        }
        else {
            bandera2=0;
            rutina_lectura_HC();
            tiempo=get_timer1();
            tiempo=nivel_ref-tiempo;//en microsegundos
            if(tiempo<0)
                tiempo=0;
            printf("AT+HTTTPARA=\"URL\", \"http://sensor-rt.000webhostapp.com/i.php?valor=\");
            delay_ms(50);
            printf("%Lu", tiempo);
            printf("\r\n");
            delay_ms(900);
            printf("AT+HTTPACTION=0\r\n"); //sube dato con metodo GET
            delay_ms(8000);
        }
    }
    parpadeo_A0();
}
}

```

## D.2. CODIGO DE SIMULACION PARA EL PIC16F877A

```

#include<16F877A.h>
#fuses XT,NOWDT
#use standard_io(B)
#use delay(clock=4M)
#use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7)

```



```

void main(){

    while(1){
        if (input(PIN_B4)==1){
            delay_ms(300);
            printf("+CGATT: 1\r\n");
            printf("OK\r\n");
        }
        if(input(PIN_B5)==1){
            delay_ms(300);
            printf("\r\n+HTTPACTION:0,555,555\r\n");
        }
        if(input(PIN_B6)==1){
            delay_ms(300);
            printf("\r\n+HTTPACTION:\r\n");
        }
        if(input(PIN_B7)==1){
            delay_ms(300);
            printf("\r\n+HTTPACTION:0,200,0\r\n"); //408 o 200
        }
    }
}

```

### D.3. CODIGO PHP DEL SCRIPT 1

```

<?php
date_default_timezone_set('America/Managua');

$val=$_GET["valor"];
$cuenta=0;
$val_random=0;

require("datos_conexion.php");

//conexion a base de datos por procedimientos
$conexion=mysqli_connect($db_host,$db_usuario,$db_contra);
if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}
mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion,"utf8"); //para que incluya los caracteres latinos

//Normalizamos $val en milímetros
$val=(int)($val*17/100);

$date1=date('Y-m-d H:i:s',time());
$consulta="INSERT INTO datos_sensor1 (VALOR,FECHA) VALUES ('$val','$date1')";
$resultados=mysqli_query($conexion,$consulta);

if($resultados==false){
    echo "Error en la consulta";
}
else{

```

```

        echo "Registros guardados con exito<br>";
    }
    //Determinar si se debe guardar $val en tabla alertas
    if($val>250) {
        $consulta2="INSERT INTO alertas (VALOR,FECHA) VALUES ('$val','$date1')";
        $resultados2=mysqli_query($conexion,$consulta2);
    }
    mysqli_close($conexion);
?>

```

#### D.4. CODIGO PHP DEL SCRIPT 2

```

<?php
$recupera=$_REQUEST["solicitar_dato"];
require("datos_conexion.php");
//conexion a base de datos por procedimientos
$conexion=mysqli_connect($db_host,$db_usuario,$db_contra);
if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}
mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion,"utf8"); //para que incluya los caracteres latinos
$consulta="SELECT * FROM datos_sensor1 ORDER BY FECHA DESC LIMIT 1";
$resultados=mysqli_query($conexion,$consulta);
if($resultados==false){
    echo "Error en la consulta";
}
else{
    if($fila = mysqli_fetch_assoc($resultados)){
        foreach ($fila as $line){
            $fila1[0]= $fila['FECHA'];
            $fila1[1]= $fila['VALOR'];
        }
        echo $fila1[0].",".$fila1[1];
    }
}
mysqli_close($conexion);
?>

```

#### D.5. CODIGO PHP DEL SCRIPT 3

```

<?php
$date=$_REQUEST["fecha"];
$date2=$_REQUEST["fecha2"];
$level=$_REQUEST["nivel"];
$level2=$_REQUEST["nivel2"];
require("datos_conexion.php");

//conexion a base de datos por procedimientos
$conexion=mysqli_connect($db_host,$db_usuario,$db_contra);
if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}

```

```

mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion,"utf8"); //para que incluya los caracteres latinos

if($date!="" and $date2==""){
    $consulta="SELECT VALOR, FECHA FROM datos_sensor1 WHERE FECHA='$date'
ORDER BY VALOR DESC LIMIT 100";
    $resultados=mysqli_query($conexion,$consulta);
    if($resultados==false){
        echo "Error en la consulta";
    }
    else{
        while($fila = mysqli_fetch_assoc($resultados)){
            foreach ($fila as $line){
                $fila1[0]= $fila['FECHA'];
                $fila1[1]= $fila['VALOR'];
            }
            echo $fila1[0]." || ".$fila1[1]."*";
        }
    }
}

else if($date!="" and $date2!="") {
    $consulta="SELECT VALOR, FECHA FROM datos_sensor1 WHERE FECHA BETWEEN
'$date' AND '$date2' ORDER BY VALOR DESC LIMIT 100";
    $resultados=mysqli_query($conexion,$consulta);
    if($resultados==false){
        echo "Error en la consulta";
    }
    else{
        while($fila = mysqli_fetch_assoc($resultados)){
            foreach ($fila as $line){
                $fila1[0]= $fila['FECHA'];
                $fila1[1]= $fila['VALOR'];
            }
            echo $fila1[0]." || ".$fila1[1]."*";
        }
    }
}

if($level!="" and $level2=="") {
    $consulta="SELECT VALOR, FECHA FROM datos_sensor1 WHERE VALOR=$level
ORDER BY FECHA DESC LIMIT 100";
    $resultados=mysqli_query($conexion,$consulta);
    if($resultados==false){
        echo "Error en la consulta";
    }
    else{
        while($fila = mysqli_fetch_assoc($resultados)){
            foreach ($fila as $line){
                $fila1[0]= $fila['FECHA'];
                $fila1[1]= $fila['VALOR'];
            }
            echo $fila1[0]." || ".$fila1[1]."*";
        }
    }
}
}

```

```

else if($level!="" and $level2!="") {
    $consulta="SELECT VALOR, FECHA FROM datos_sensor1 WHERE VALOR BETWEEN
$level AND $level2 ORDER BY VALOR DESC LIMIT 100";
    $resultados=mysqli_query($conexion,$consulta);
    if($resultados==false){
        echo "Error en la consulta";
    }
    else{
        while($fila = mysqli_fetch_assoc($resultados)){
            foreach ($fila as $line){
                $fila1[0]= $fila['FECHA'];
                $fila1[1]= $fila['VALOR'];
            }
            echo $fila1[0]." || ".$fila1[1]."*";
        }
    }
}
mysqli_close($conexion);
?>

```

## D.6. CODIGO PHP DEL SCRIPT 4

```

<?php
$recupera=$_REQUEST["alerta"];
require("datos_conexion.php");

//conexion a base de datos por procedimientos
$conexion=mysqli_connect($db_host,$db_usuario,$db_contra);
if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}
mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BBDD");
mysqli_set_charset($conexion,"utf8"); //para que incluya los caracteres latinos
$consulta="SELECT * FROM alertas ORDER BY VALOR DESC LIMIT 100";
$resultados=mysqli_query($conexion,$consulta);
if($resultados==false){
    echo "Error en la consulta";
}
else{
    while($fila = mysqli_fetch_assoc($resultados)){
        foreach ($fila as $line){
            $fila1[0]= $fila['FECHA'];
            $fila1[1]= $fila['VALOR'];
        }
        echo $fila1[0]." || ".$fila1[1]."*";
    }
}
mysqli_close($conexion);
?>

```

## ANEXO E: CODIGOS DE BLOQUE DE MIT APP INVENTOR

### E.1. CODIGO DE SCREEN PRINCIPAL

Este código de bloque es demasiado grande como para colocarlo todo en una imagen en una sola página, así que se dividirá en imágenes conteniendo parte del código principal. Al final no importa en qué orden se coloquen en el editor de bloques, solamente se deben colocar todos juntos dentro del editor.

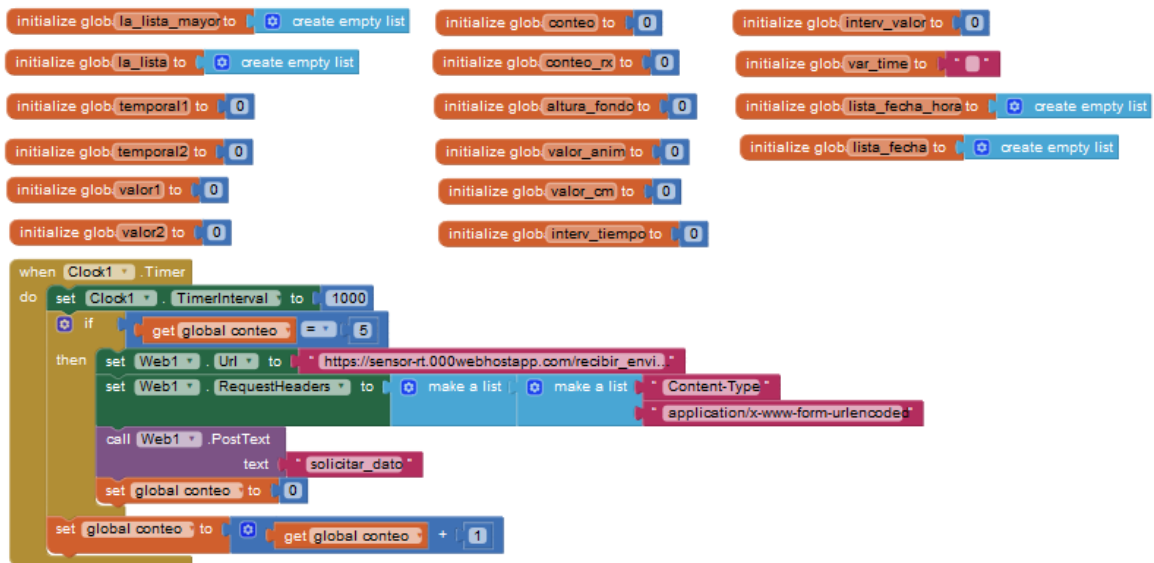


Figura E.1. Definición de variables y rutina del temporizador.

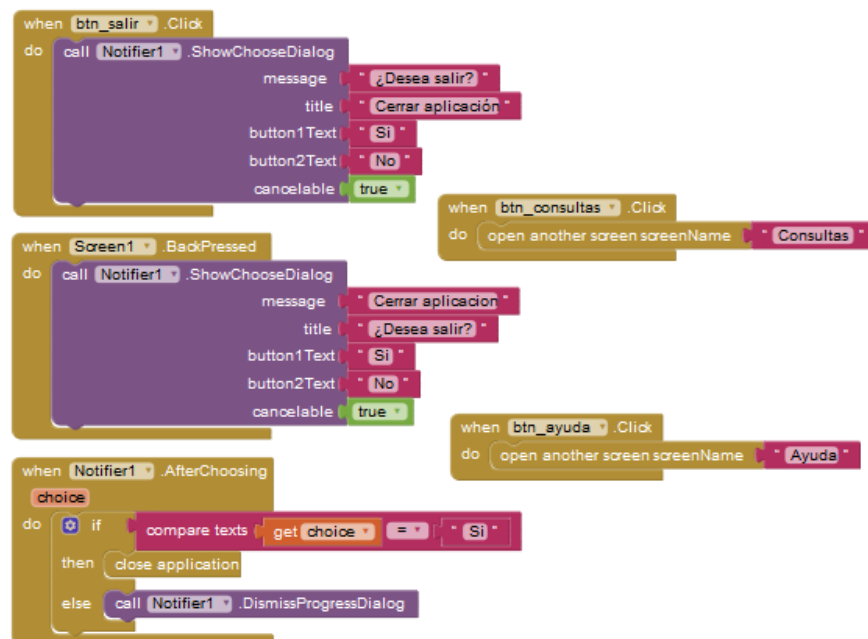


Figura E.2. Bloques asociados al cierre de la App y de enlace a otras screens.

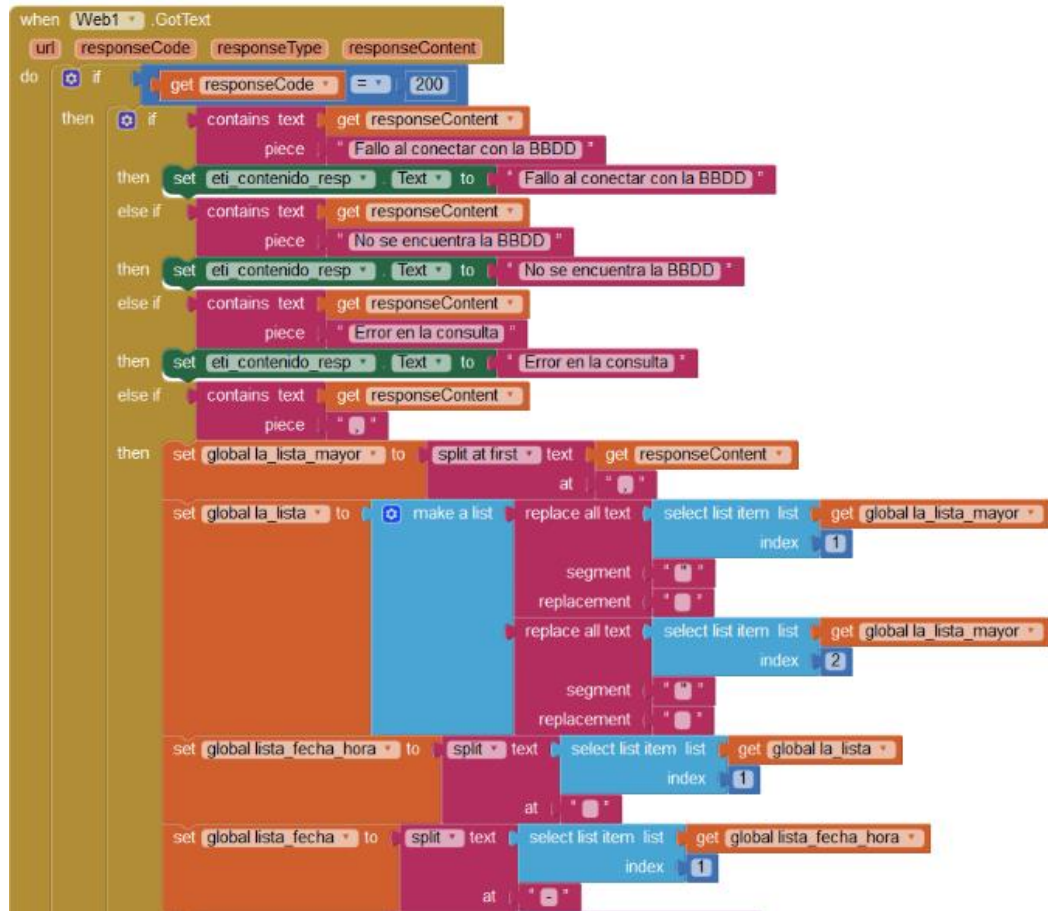


Figura E.3. Bloque principal (parte 1/3)

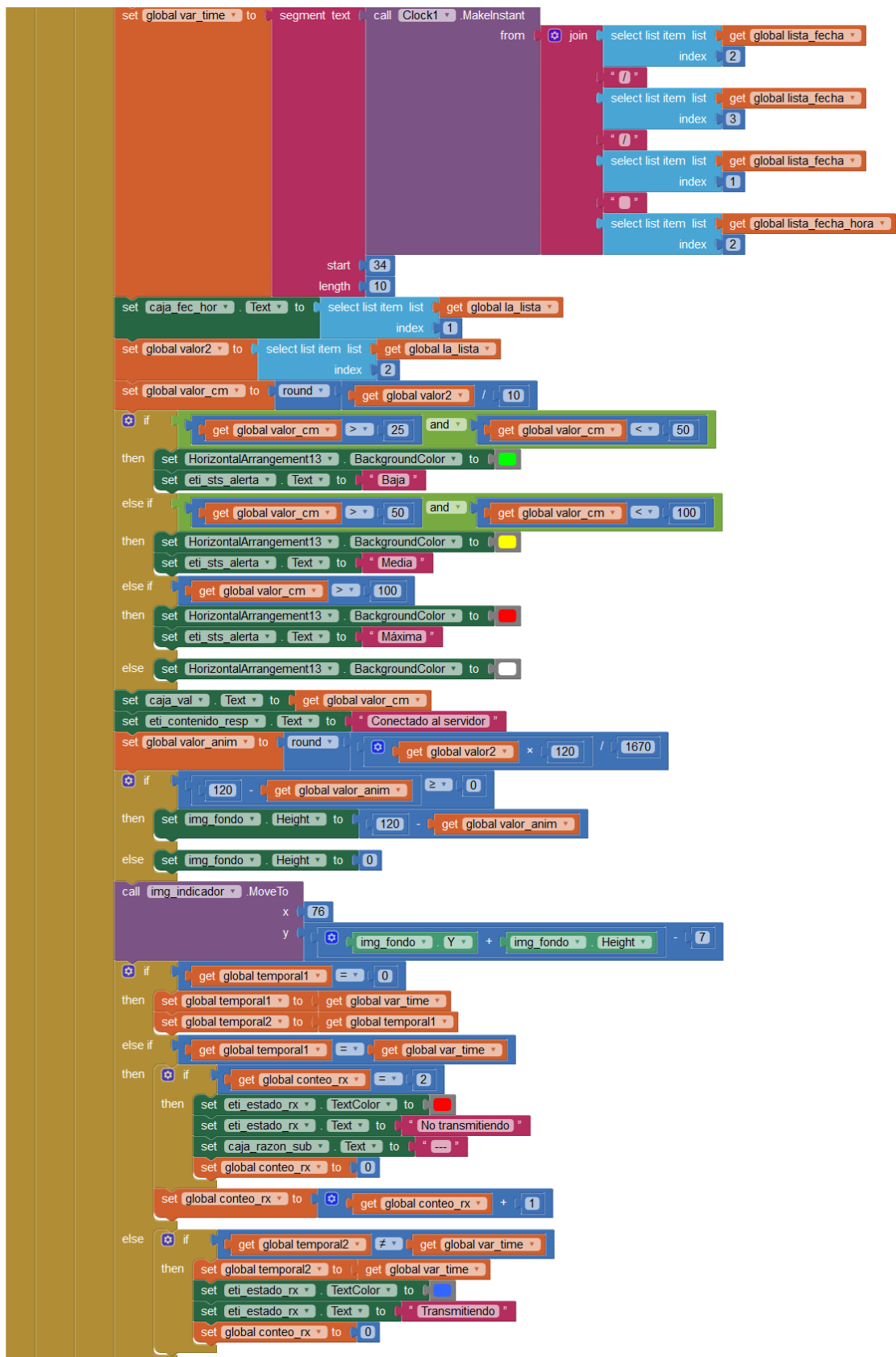
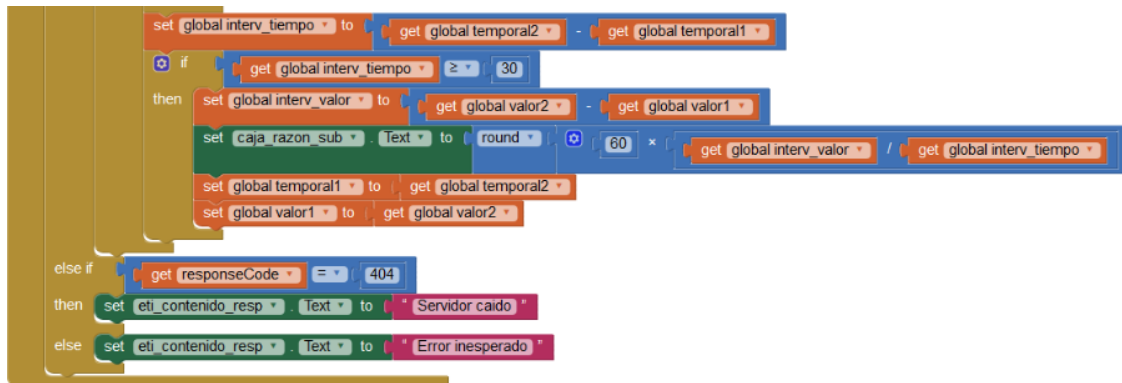


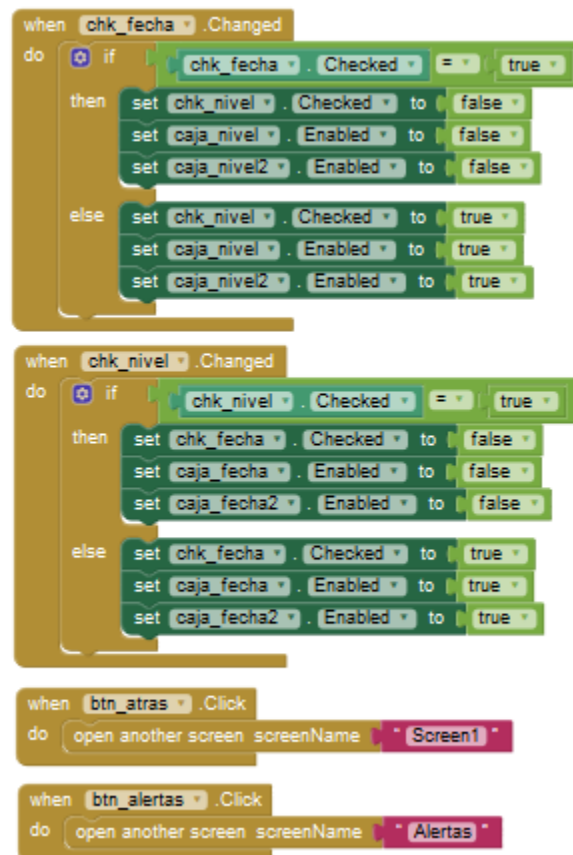
Figura E.4. Bloque principal (parte 2/3)



**Figura E.5.** Bloque principal (parte 3/3)

## E.2. CODIGO DE SCREEN CONSULTAS

Dado el considerable tamaño del código de bloques, este se dividirá en partes para ser mostrados.



**Figura E.6.** Bloques para casillas de verificación y botones de enlace a otras screens



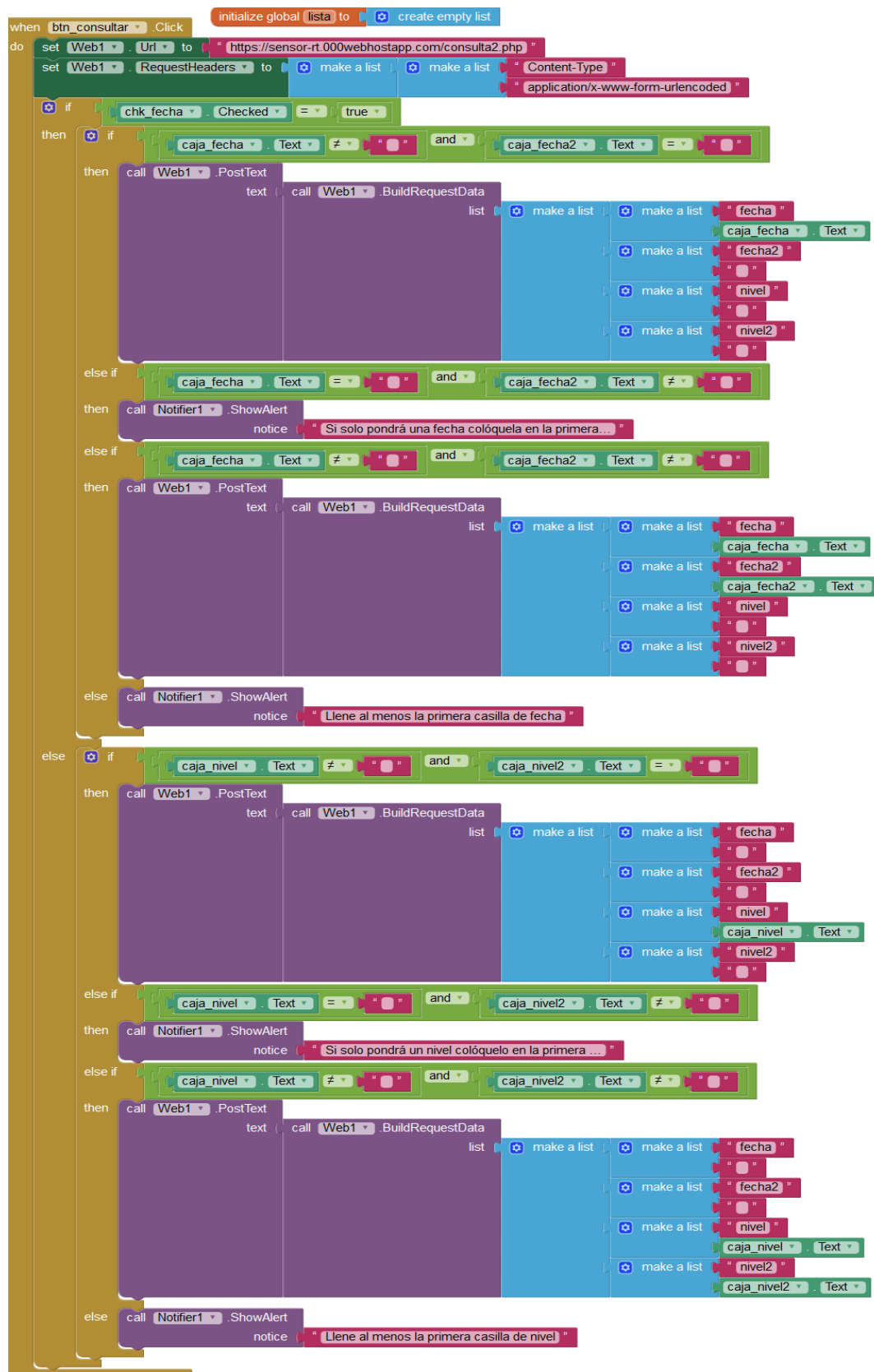
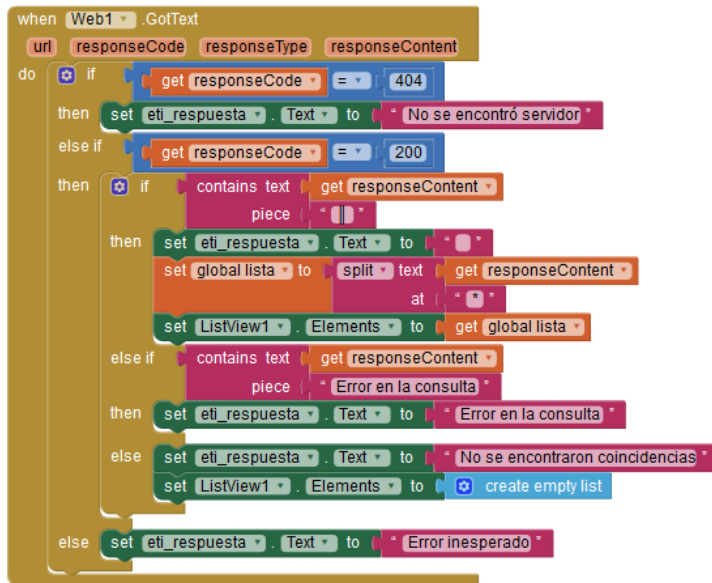
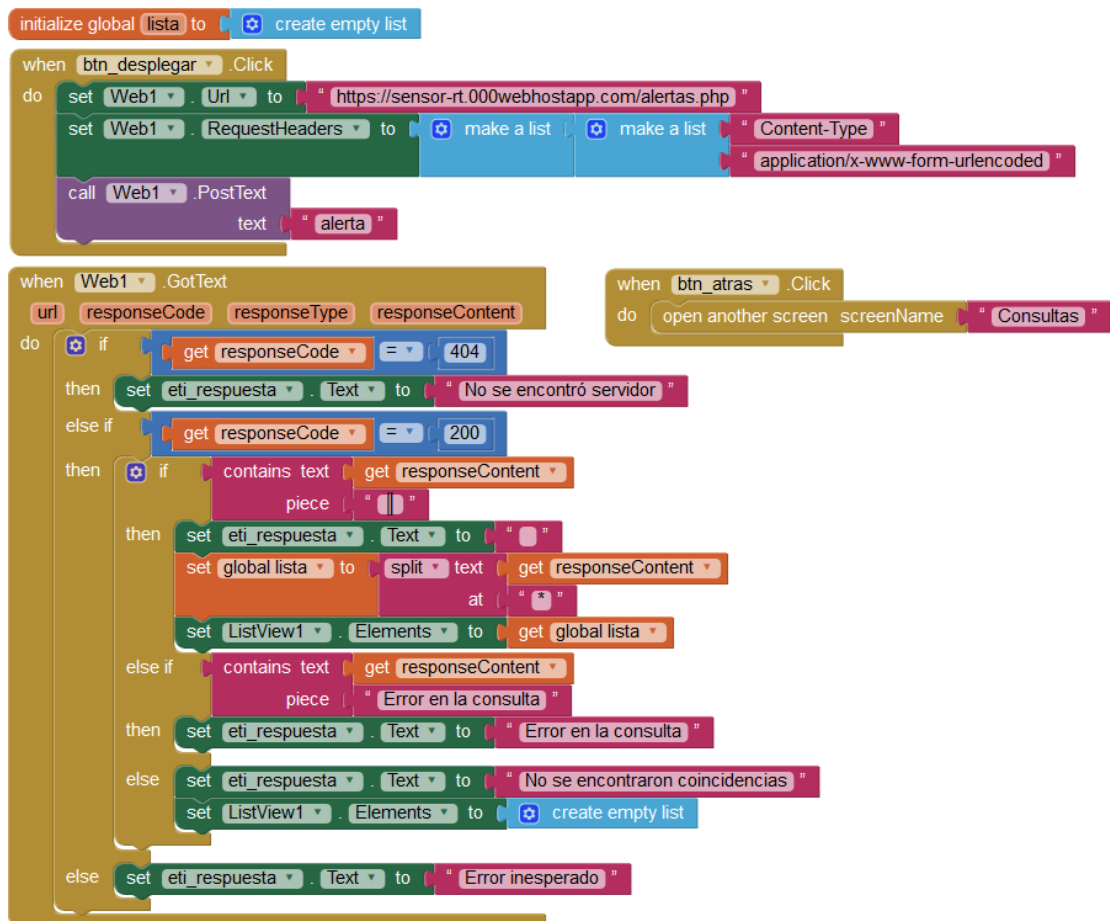


Figura E.7. Bloque principal

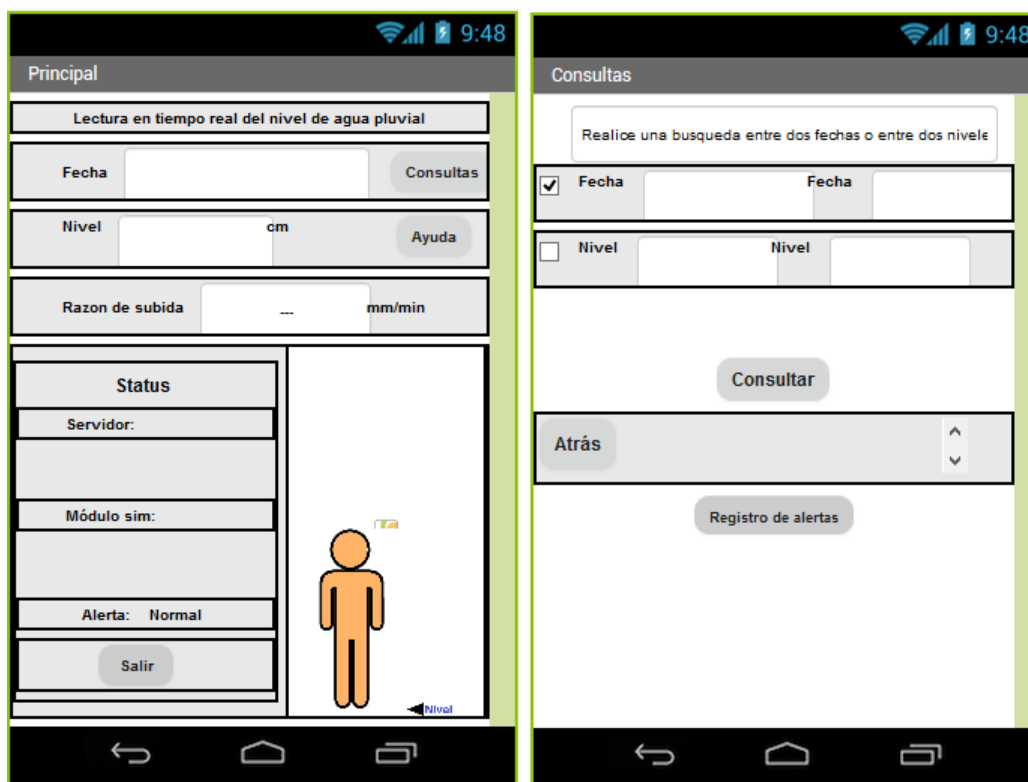


**Figura E.8.** Bloque de control de respuesta del servidor

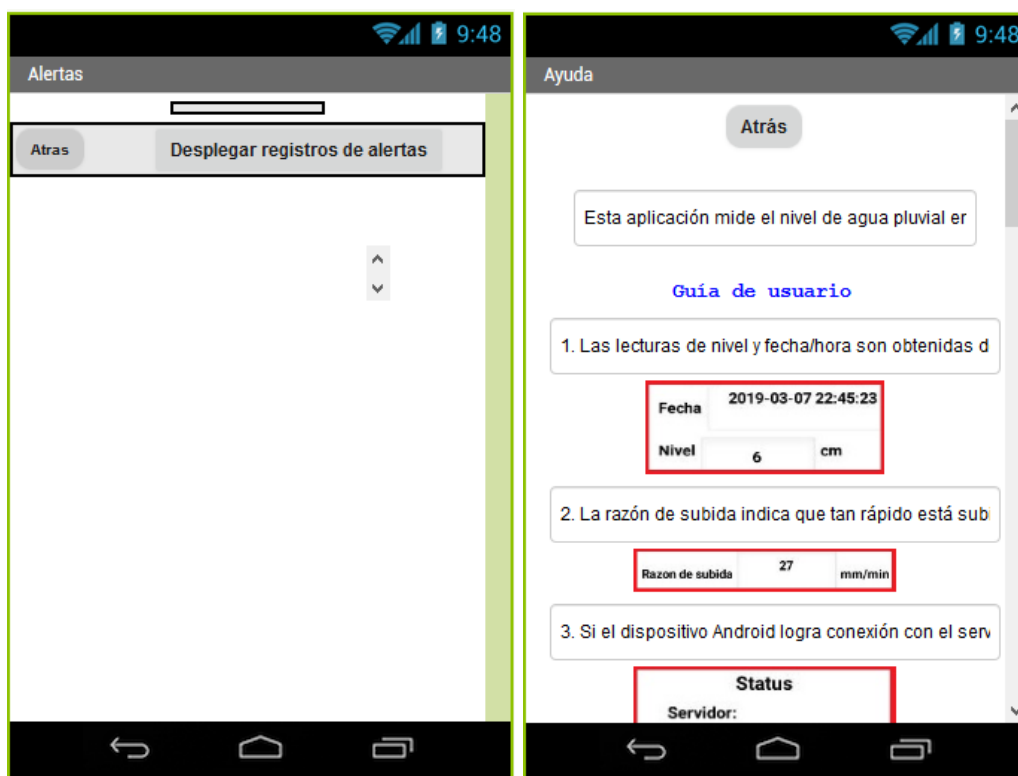
### E.3. CODIGO DE SCREEN ALERTAS



**Figura E.9.** Código de screen alertas



**Figura E.10.** Screen principal (izq.) y screen consultas (der.). Vista de diseño

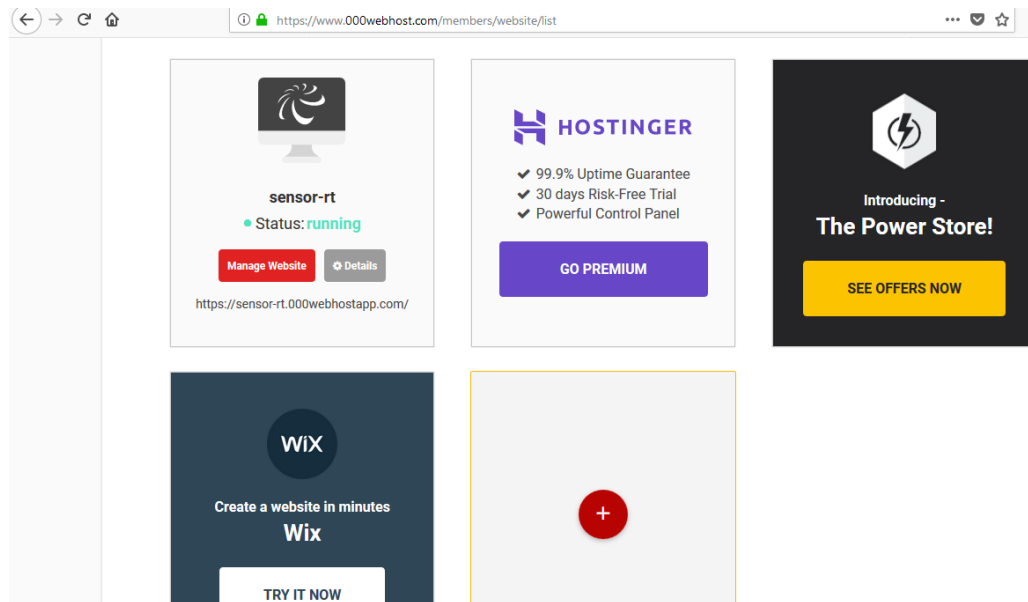


**Figura E.11.** Screen alertas (izq.) y screen ayuda (der.). Vista de diseño.

## ANEXO F: CREACION DEL SERVIDOR WEB Y LA BASE DE DATOS

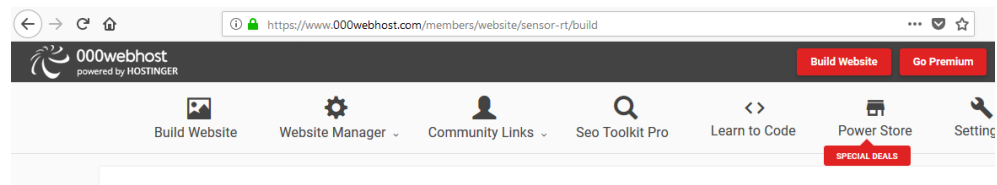
### F.1. PASOS PARA MONTAR EL SERVIDOR CON UN WEB HOSTING

1. En el navegador web dirigirse a <https://www.000webhost.com/>
2. Crear una cuenta.
3. Crear un nuevo sitio web dando click en el signo “+” como se ve en la figura F.1.



**Figura F.1.** Pantalla de creación de sitio web.

4. En la venta emergente colocar el nombre del sitio web y la contraseña con la cual se accederá para futuras configuraciones.
5. Hacer click y en “Build Website” como se ve en la figura F.2.



**Figura F.2.** Pantalla de selección de Build Website

6. Desplazase sobre la página hacia abajo y hacer click en “UPLOAD NOW”. Se abrirá una ventana nueva en nuestro navegador.

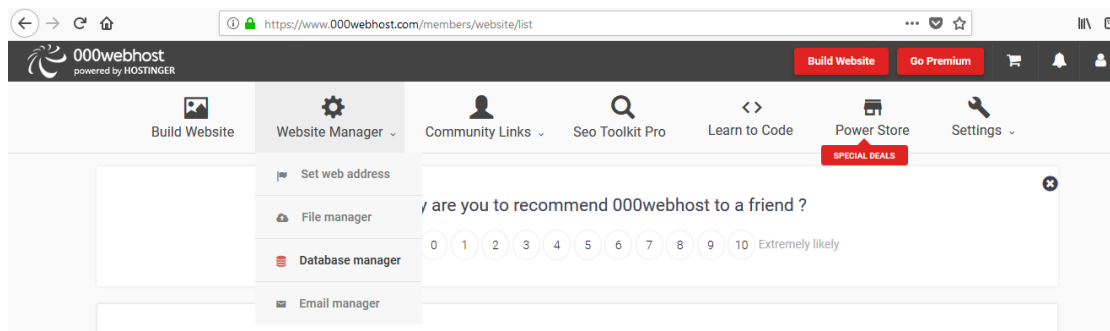
7. Finalmente se pueden subir todos los archivos php requeridos dando click en la opción “Upload Files”, como se muestra en la figura F.3.



**Figura F.3.** Interfaz de selección de archivos.

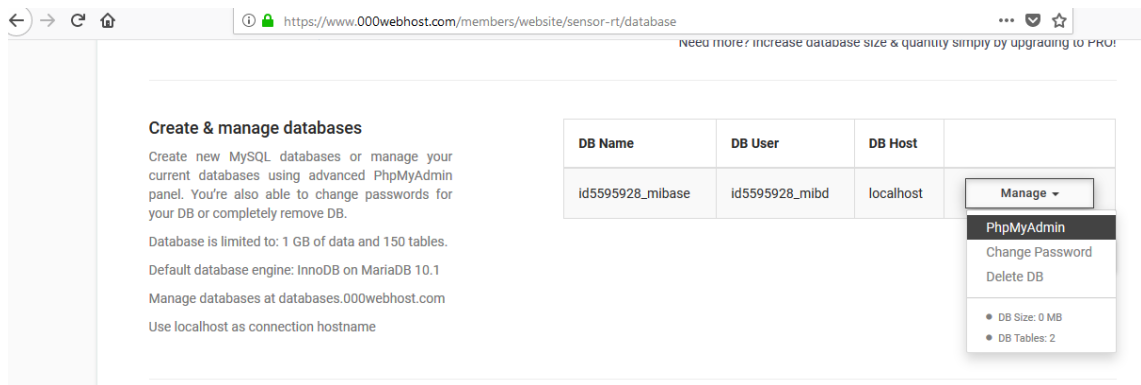
## F.2. PASOS PARA LA CREACION DE LA BASE DE DATOS

1. Tener una cuenta previamente creada en <https://www.000webhost.com/> y acceder a esta.
2. Dirigirse a “Website Manager” y de la lista desplegable seleccionar “Database Manager” como se muestra en la figura F.4.



**Figura F.4.** Pantalla de selección de administrador de bases de datos

3. Desplazar la página hacia abajo y hacer click en “New Database”.
4. En la venta emergente seleccionar un nombre para la BD, un nombre de usuario y una contraseña. Luego hacer click en “Create” para finalizar.
5. Una vez creada la BD, para ingresar a esta con phpMyAdmin, ir a “Website Manager”->“Database Manager”. Luego hacer click en “Manage” y seleccionar phpMyAdmin, como se observa en la figura F.5.



**Figura F.5.** Pantalla de selección de BD.

## ANEXO G: ENCUESTA

### Encuesta

1. **¿Tiene usted conocimiento de que existen puntos críticos de inundación pluvial en Managua?**
  - ☐ Si
  - ☐ No
2. **¿En que medio te has enterado de esto?**
  - ☐ Periódicos
  - ☐ TV
  - ☐ Otro medio
3. **¿A quién te imaginas que le puede interesar conocer el nivel de inundación en tiempo real en una zona?**
  - ☐ Alcaldía
  - ☐ Sinapred
  - ☐ Solamente los afectados.
  - ☐ Público en general
4. **¿En tu opinión crees que las brigadas de rescate responden rápidamente en caso que personas que se encuentren en peligro durante una inundación?**
  - ☐ Si
  - ☐ No
5. **¿Qué te parecería tener a disposición una aplicación Android que permita conocer el nivel de agua en esas zonas durante una lluvia torrencial?**
  - ☐ Interesante
  - ☐ Poco interesante
  - ☐ Nada interesante
6. **¿Crees que esta herramienta podría servir a las autoridades involucradas en rescate a desplegar con más rapidez sus brigadas?**
  - ☐ Si
  - ☐ No
7. **¿Cómo piensas que debe ser la aplicación para fines de monitoreo de nivel de inundación?**
  - ☐ De fácil uso de manera que cualquier usuario pueda entender
  - ☐ Debe estar diseñada solo para uso de expertos.
8. **¿Tienes conocimiento de la existencia de una aplicación Android de monitoreo en tiempo real al menos aquí en Nicaragua? Si la respuesta es sí, ¿recuerdas el nombre de esa App?**
  - ☐ Si
  - ☐ No

## Resultados de la encuesta

Por precaución la encuesta fue realizada en línea ya que no se conoce el nivel de seguridad del punto crítico considerado. Esta encuesta se realizó a través de redes sociales en las cuales se consultó a una muestra de 30 personas del municipio de Managua a cerca de los puntos críticos de inundación y sobre el uso de una App para el monitoreo de inundación de los puntos críticos en general.

**Tabla G.1**

*Resultados de encuesta*

Pregunta	Respuesta			
1	Si : 28	No: 2		
2	Periódico: 6	TV: 24	Otro medio: 0	
3	Alcaldía: 6	Sinapred: 11	Solo afectados: 4	Publico general: 9
4	Si: 14	No: 16		
5	Interesante: 17	Poco interesante: 10	Nada interesante: 3	
6	Si: 27	No: 3		
7	Si: 30	No: 0		
8	Si: 0	No: 30		

**Nota.** Encuesta realizada a 30 personas del municipio de Managua, sin tomar en cuenta el género.



## ANEXO H: PRESUPUESTO

**Tabla H.1**

*Presupuesto general*

No.	Item	Descripción	Precio (C\$)	Costo de envío	Cantidad	Total (C\$)
1	Shield GSM/GPRS	Módulo de comunicación y transmisión	1015.36	0	1	1015.36
2	Caja plexo	Panel de conexión eléctrica 220x170x140	1052.43	0	1	1052.43
3	Modulo solar	Modulo fotovoltaico 30W 12V	873.18	0	1	873.18
4	Batería	Batería sellada Ritar 12V 9Ah	751.48	0	1	751.48
5	Controlador	Controlador de carga 12V/24V 10A	315.8	157.9	1	473.7
6	PIC16F688 I/P	Integrado de control	157.9	157.9	1	315.8
7	MAX232	Driver/Receptor TTL-RS232	65	0	1	65
8	HC-SR05	Sensor ultrasónico 5V	160	0	1	160
9	Estructura metálica	Montaje de aluminio del sistema	200	0	1	200
10	Placa	Placa para PCB 250x200	100	0	1	100
11	Resistor	Resistor de 4.7 kΩ	3	0	2	6
12	Resistor	Resistor de 10 kΩ	3	0	2	6
13	LED	LED común de 3mm	5	0	2	10
14	Capacitor	Capacitor electrolítico de 10 µF 50V	5	0	4	20
15	Capacitor	Capacitor de tantalio 4.7 µF 16V	40	0	1	40
16	Capacitor	Capacitor de poliéster	20	0	1	20
17	2N2222	Transistor NPN	20	0	1	20
18	Tornillos	Goloso punta fina 1/2"	3.46	0	8	27.68
19	Tuerca	Tuerca hexagonal 6/32"	3.46	0	12	41.52
20	Perno	Perno 6/32x1-1/2"	3.46	0	10	34.6
21	Terminal	Terminal de bloque x2	20	0	4	80
22	Terminal	Terminal de gaveta	7	0	2	14
					Total	5326.75

**Nota.** Algunos artículos fueron comprados en dólares, por lo que se hizo una conversión a córdobas según el tipo de cambio en la fecha que se compró,

## ANEXO I: OTROS DOCUMENTOS



**Universidad Nacional de Ingeniería**  
**Recinto Universitario "Simón Bolívar"**  
**Facultad de Electrotecnia y Computación**  
Vice-Deanatura

Managua, 08 de Octubre del 2018.

**Ing. Rodolfo Martin Villachica**  
**Director General de proyectos de la alcaldía de Managua**  
**Sus manos**

Estimado Ingeniero Villachica:

La siguiente misiva es para solicitar de la manera más atenta la colaboración que le pueda brindar al estudiante de la carrera de Ingeniería en Electrónica de la Facultad de Electrotecnia y Computación de la Universidad Nacional de Ingeniería. Detallo datos de los estudiantes:

Nombre	Carnet
Jorge Noel Real Martínez	2013-61894

El apoyo necesario para que el estudiante realice su trabajo monográfico, para eso requiere el estudiante información referida la ubicación de los puntos críticos de inundación pluvial en el distrito V del municipio de Managua y datos de las estaciones Hidropluviotelemetricas instaladas ahí.

No omito manifestar que esta actividad se realiza con el objetivo de complementar la formación de nuestros estudiantes, siendo consistente con la excelencia académica, premisa fundamental de nuestra Alma Mater.

De antemano agradeciendo su apoyo.

Atentamente,



**Ing. Augusto César Palacios Rodríguez**  
**Vice – Decano**  
**Facultad de Electrotecnia y Computación.**  
**Teléfono: 2267-3348**



Cc.

Ing. Maritza Maradiaga  
Archivo

Dir. Formulación, Seguimiento y Evaluación de proyecto

9-10-18.